DOCUMENT RESUME

ED 126 561 95 EA 008 502

AUTHOR	Belt, Sidney L.; Spuck, Dennis W.
TITLE	Design Specifications for the Generalized Wisconsin System for Instructional Management (WIS-SIM). Working Paper No. 133 (Part 2 of 2 Parts).
INSTITUTION	Wisconsin Univ., Madison. Research and Development Center for Cognitive Learning.
SPONS AGENCY	National Inst. of Education (DHEW), Washington, D.C.
REPORT NO	WRDCCL-WP-133-Part-2
PUB DATE	Sep 75
CONTRACT	NE-C-00-3-0065
NOTE	89p.; Report from the Project on Computer Applications in Individually Guided Education
EDRS PRICE	MF-$0.83 HC-$4.67 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Oriented Programs; *Computer Programs; Elementary Education; *Individualized Instruction; *Management Information Systems; Models; Program Design; *Program Development; *Program Evaluation
IDENTIFIERS	*Wisconsin System of Instructional Management; WIS SI

ABSTRACT

This publication is part 2 of a two-part paper that presents design specifications for the Wisconsin System of Instructional Management (WIS-SIM), a computerized management system designed for instructional programs that are compatible with the Individually Guided Education model. WIS-SIM incorporates processes for placement testing, performance profiling, and specifying performance specifications to aid in diagnosing and identifying students' instructional needs. Once instructional needs have been identified, WIS-SIM aids in guiding the instructional process and selecting appropriate instructional activities, and in evaluating the success of the instructional program. This part of the paper discusses ways of adapting WIS-SIM for use with a variety of available computer configurations and outlines plans for further development and evaluation of the system. The appendix presents detailed information on coding data and designing programs for particular applications of WIS-SIM. (Author/JG)

Working Paper No. 133 (Part 2 of 2 Parts)

DESIGN SPECIFICATIONS FOR THE GENERALIZED WISCONSIN SYSTEM FOR INSTRUCTIONAL MANAGEMENT (WIS-SIM)

by

Sidney L. Belt and Dennis W. Spuck

with

Nancy Ching-Ching Chan, Carol L. Dagnon, Bruce G. Douglas,
Brian F. Lawrence, James R. McNamara, and Stephen P. Owen

Report from the Project on
Computer Applications in Individually Guided Education

Dennis W. Spuck
Principal Investigator

Wisconsin Research and Development
Center for Cognitive Learning
The University of Wisconsin
Madison, Wisconsin

September 1975

Published by the Wisconsin Research and Development Center for Cognitive Learning, supported in part as a research and development center by funds from the National Institute of Education, Department of Health, Education, and Welfare. The opinions expressed herein do not necessarily reflect the position or policy of the National Institute of Education and no official endorsement by that agency should be inferred.

Center Contract No. NE-C-00-3-0065

WISCONSIN RESEARCH AND DEVELOPMENT CENTER FOR COGNITIVE LEARNING

MISSION

The mission of the Wisconsin Research and Development Center for Cognitive Learning is to help learners develop as rapidly and effectively as possible their potential as human beings and as contributing members of society. The R&D Center is striving to fulfill this goal by

- conducting research to discover more about how children learn
- developing improved instructional strategies, processes and materials for school administrators, teachers, and children, and
- offering assistance to educators and citizens which will help transfer the outcomes of research and development into practice

PROGRAM

The activities of the Wisconsin R&D Center are organized around one unifying theme, Individually Guided Education.

FUNDING

The Wisconsin R&D Center is supported with funds from the National Institute of Education; the Bureau of Education for the Handicapped, U.S. Office of Education; and the University of Wisconsin.

4

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Lawrence E. Winkler for his contribution to the design of the interactive programming concepts, to Sara West for her critical reading of the document, and to Laura Middleton for her numerous retypings of the various sections. Thanks are also expressed to Sharon Stevens of the Media Services Section of the Wisconsin Research and Development Center for her contribution to the editing and production of the final document.

TABLE OF CONTENTS

6

TABLE OF CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (cont.)

LIST OF FIGURES (cont.)

ABSTRACT

This paper deals with design specifications of a generalized system of computer management for instructional programs which are compatible with the model of Individually Guided Education (IGE). Computer managed instruction (CMI) seeks to facilitate processing information and supplying this information at appropriate times and places so that it is directly applicable to instructional decision making.

A model for the generalized WIS-SIM is developed. This model incorporates the processes of testing, performance profiling, specifying performance expectations, diagnosing and identifying instructional needs, guiding the instructional process and selecting appropriate educational experiences and settings, instructing, and evaluating the instructional program.

The information flow in the generalized system is discussed and the related data bases are specified. The developmental schedule is given and an approach to evaluation is outlined.

V

ADAPTING TO AVAILABLE COMPUTER CONFIGURATIONS

A major design goal of WIS-SIM is to make a CMI system available to a large number of IGE schools (Belt & Spuck, 1974). One response to this goal would be to develop a system which could be implemented on a computer configuration that many schools could afford. Such an approach is not viable in that it seems unlikely that many school systems will be able to acquire computers for the sole purpose of managing instructional programs. A more promising approach is to design a system that is compatible with computer configurations which are available, or likely to become available, to schools. Computing capability may be available to schools from the following four sources:

1. The school district
2. Cooperative Regional Educational organizations
3. Universities
4. Commercial service bureaus

The computer configurations available from these sources cover a broad spectrum from batch administrative data processing systems to on-line mini-computer systems to large-scale multiprocessing computer systems which can concurrently support interactive terminals, on-line terminals, and batch data processing.

Administrative data processing systems found in school districts and Cooperative Regional Educational organizations generally operate in the batch mode with little or no teleprocessing capability. School districts and Cooperative Regional Educational organizations are also acquiring on-line minicomputer systems for interactive instructional data processing. These systems are being used to teach programming to students and to support gaming and simulation exercises, drill and practice, and tutorial Computer Aided Instruction (CAI). Most of these minicomputer systems do not have the mass storage capability required to support CMI application. However, there seems to be a trend toward upgrading these systems or replacing them with minicomputer systems having larger mass storage capabilities. These systems, therefore, hold some potential for future support of computer managed instruction.

It is difficult to characterize the typical university configuration. A large university may have a large-scale, multiprocessing configuration which can support batch and interactive modes of operation concurrently. Or, the university may have a batch system which is used for both academic and administrative data processing. Frequently, universities have minicomputer configurations which were acquired primarily to provide an interactive problem-solving capability on campus; some campuses have also made this capability available to a larger community of users.

Commercial service bureaus cover a broad spectrum of capabilities: Depending on the particular service bureau, batch processing may be provided on a small-, medium-, or large-scale system. For those service bureaus that do provide on-line or interactive service, a large-scale computer is generally utilized. Service bureaus with on-line or interactive capability usually cater to a regional or national clientele, while bureaus serving local clientele generally have only batch capability. When using a service bureau, it would be well for schools to consider having their data processed during second or third shift operations in order to take advantage of lower rates.

Interactive and on-line systems have great appeal in the CMI application because turnaround time, the time required to enter data and receive reports, is minimized. However, batch implementation will continue to be given significant attention in the design of WIS-SIM because it is the level of service that is most available to schools. Implications for implementing WIS-SIM on interactive, on-line, and batch configurations will be discussed later in this chapter.

SOURCE DATA COLLECTION

An optional feature of all three modes of operation (on-line, interactive, and batch) is "source data collection." Ideally, data generated as a result of normal classroom activity would be recorded in a form which can be directly entered into the computer.

~~Source~~ data collection is related to the major design goal of minimizing demands on teachers to learn the system and associated tasks which are different from normal classroom activities. The aspect of source data collection which is usually emphasized is the ability to omit the intermediate step of converting data into machine-readable form. This results in a saving in keypunching and a decrease in turnaround time. Source data collection is also appealing because it permits the implementation of an on-line system with minimal requirements ~~for school staff~~ involvement and training. Procedures based upon a source data collection concept utilizing optical mark-sense technology have been developed and promise to minimally disrupt normal classroom activity.

Although a school terminal would have a full alpha-numeric keyboard, the design goal, in cases where the school terminal also has an optical mark-sense capability, is to have as many of the inputs as feasible made through the mark-sense reader. The keyboard in such cases will be used mainly for updating small amounts of data and for newly emerging functions for which there has not been sufficient time to develop forms. The mark-sense formats are being engineered to make minimal demands on the teacher in terms of both the number of entries required per transaction and the amount of training required to become proficient in their use.

The extent to which computer configurations have mark-sense capabilities will vary considerably, including no mark-sense capability at all. Some batch systems will have a mark-sense reader at the central site. In the case of batch systems not equipped with mark-sense capabilities, data entry will generally be by keypunch at the central site.

On-line and interactive systems may have mark-sense capability either at the central site or as part of the school terminal. When mark-sense capability is located at the central site, mark-sense forms will be sent from the schools to the central site for processing. On-line and interactive systems without a mark-sense capability could either have all data inputs made through the terminal keyboard at the school or utilize some combination of terminal keyboard entry and keypunch at the central site.

A teleprinter is specified for the school terminal instead of a video tube display since hard copy is essential in many of the reports generated. Even if both a video tube display and a teleprinter could be made available at reasonable cost, there does not seem to be any generalized advantage of a video tube display for this application.

The teleprinter specified is an impact printer which can generate spirit (Ditto) masters of instructional materials. It also has an upper- and lower-case character set which will enhance readability of curriculum materials, especially for elementary school children. The teleprinter which has been acquired for the pilot and field tests can print up to 120 columns per line. This capability will provide flexibility in printing administrative reports. However, the software which is being developed will permit a dynamic restructuring of report formats down to 72 columns per line. This will ensure compatibility with the ubiquitous model 33 teletype.

MODES OF OPERATION: BATCH, ON-LINE, AND INTERACTIVE

The most significant characteristic of batch systems, in terms of CMI, is the absence of school terminals. Thus, reports generated at the computer center are delivered to the school by mail or by courier service. Requests for reports based on information the computer already has may be telephoned into the computer center; but voluminous input from the schools must be conveyed by mail or courier. In either case, requests for reports and input data are checked by a data clerk, processed, and mailed or sent by courier to the requesting school. A courier service can appreciably reduce the time required for mail delivery, but may cost more than is desirable. It is sometimes possible to use school delivery services that already exist. Turnaround time ranges from one to three days, depending on the method of conveyence. In the batch mode, data is entered into the system by keypunching or, when available, by mark-sense readers. Figure 31 shows the general batch system information flow and it illustrates the mark-sense reader option.

Both the on-line mode and the interactive mode require a computer configuration which includes a teleprocessing capability. Two significant distinctions between the on-line mode and the interactive mode are that an on-line operation can be supported with fewer computer resources and that

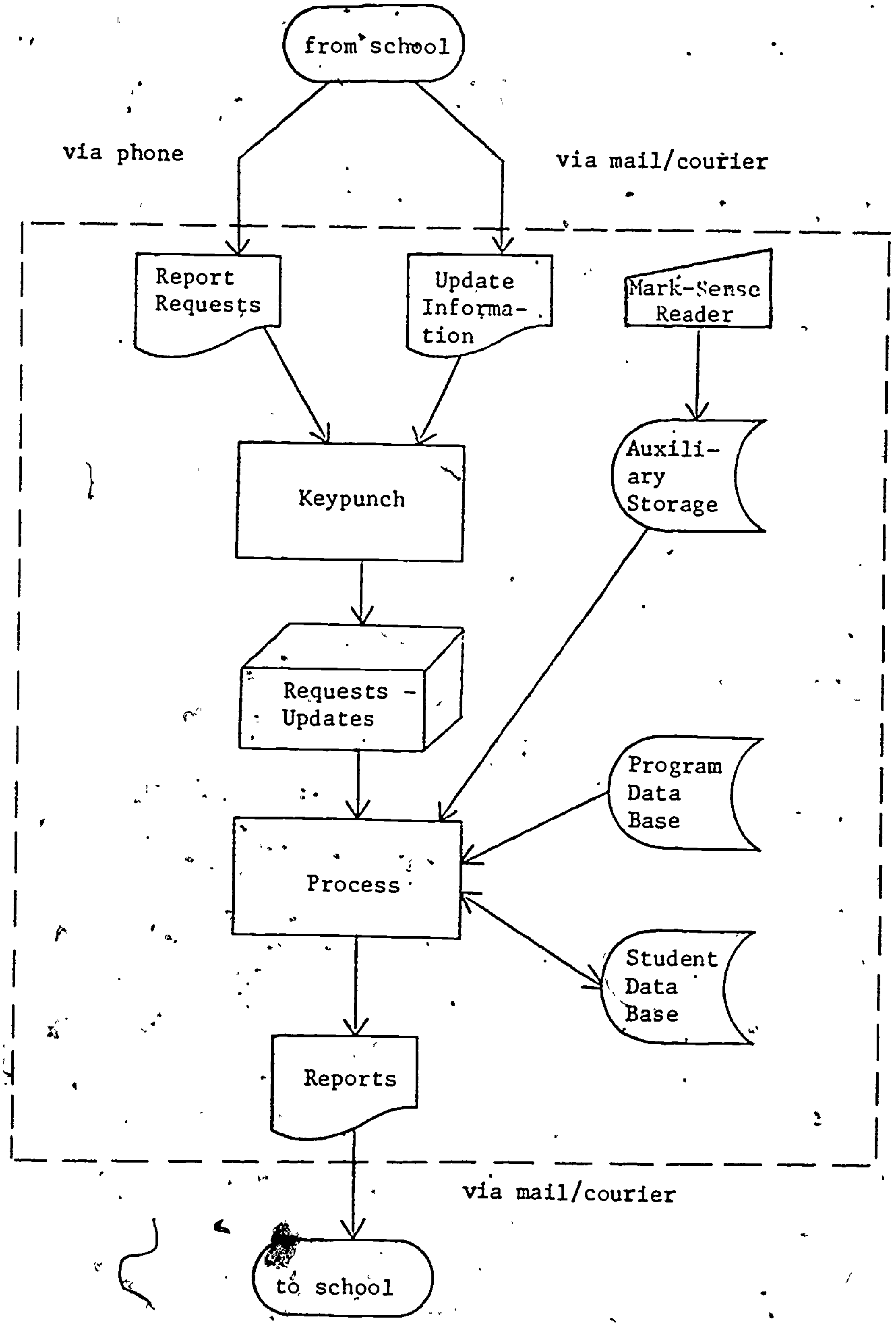


Figure 31. Generalized information flow for batch computer configurations.

Note: Dotted lines enclose central computer facility.

the turnaround time in the interactive mode is measured in seconds and minutes, while turnaround time in the on-line mode may require several hours. This latter time span, however, is still more than adequate for most if not all computer managed instruction applications.

A teleprocessing capability (that is, transmission of computer data over phone lines) is not currently characteristic of many school computer configurations, although it is very probably the wave of the future.. A teleprocessing capability can evolve in the schools in a number of ways: by upgrading or replacing current batch systems, by acquiring minicomputer systems that have this capability, or by acquiring terminal equipment to interface with large time-sharing computer systems or networks.

Response time in systems with teleprocessing capability can vary widely. Interactive systems are characteristically faster than on-line systems; they must be configured so that it appears to the user that he has the undivided attention of the computer. On-line systems with teleprocessing capability, on the other hand, may accept data, store it for varying lengths of time, and process it when convenient. This feature allows many degrees of freedom in the design of on-line systems and procedures which, when compared with that needed by interactive systems, can considerably decrease the computer resources required. For example, a parsimonious but viable implementation of an on-line system would utilize the teleprocessing capability only for transmitting reports from the computer site to the school. Requests for reports would be telephoned from the schools to a data clerk at the computer center. Updates to the data base would be sent from the schools to the computer center by mail or courier and would be keypunched or entered through a mark-sense reader at the site. On-line operation requires minimal secondary storage in that the school files need not be on-line until that school's reports are scheduled to be processed. When the reports are ready, they are then transmitted to the school's terminal. Figure 32 depicts this on-line mode of operation where the on-line teleprocessing capability is employed in only one direction--from computer center to school.

More typical on-line systems will use their telecommunications capability for transmission of data in both directions. Figure 33 shows such an on-line configuration where inputs from the school terminal (keyboard teleprinter with or without a mark-sense reader) are stored in an auxiliary storage device until the school is scheduled to have its data processed.

Since data from the schools are entered directly into an auxiliary storage device, the computer does not check the data for errors as they are being transmitted and stored. Immediate error-checking capability is characteristic of interactive systems. Because an on-line system lacks this immediate error-checking capability, designers and implementors of on-line systems must give careful attention to requirements of staff training and human engineering of procedures.

The on-line mode has advantages over the batch mode in that the delivery of reports from the computer center does not depend entirely on mail or a courier service and thus the turnaround time is reduced from several days to many minutes or a few hours, depending upon the computer facility's workload.

In the interactive mode, all requests for reports and updates are submitted to the computer through the school terminal. However, even in interactive systems, the system designer has an opportunity to trade off response time against computer resource utilization. Such a trade off is highly dependent upon how the school's data base is maintained at the

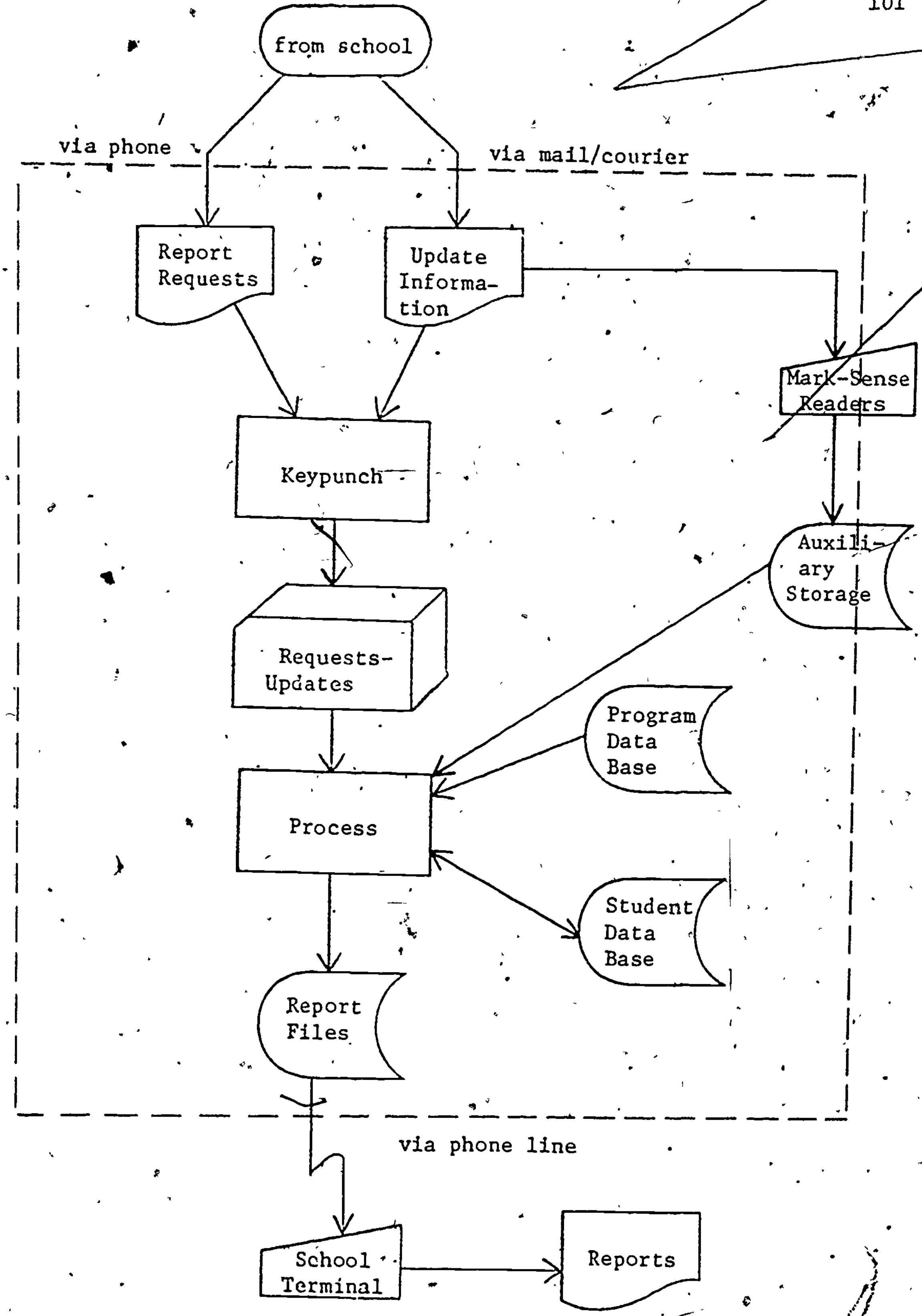


Figure 32. On-line system where telecommunications is employed in only one direction (from computer center to school).

Note: Dotted lines enclose central computer facility.

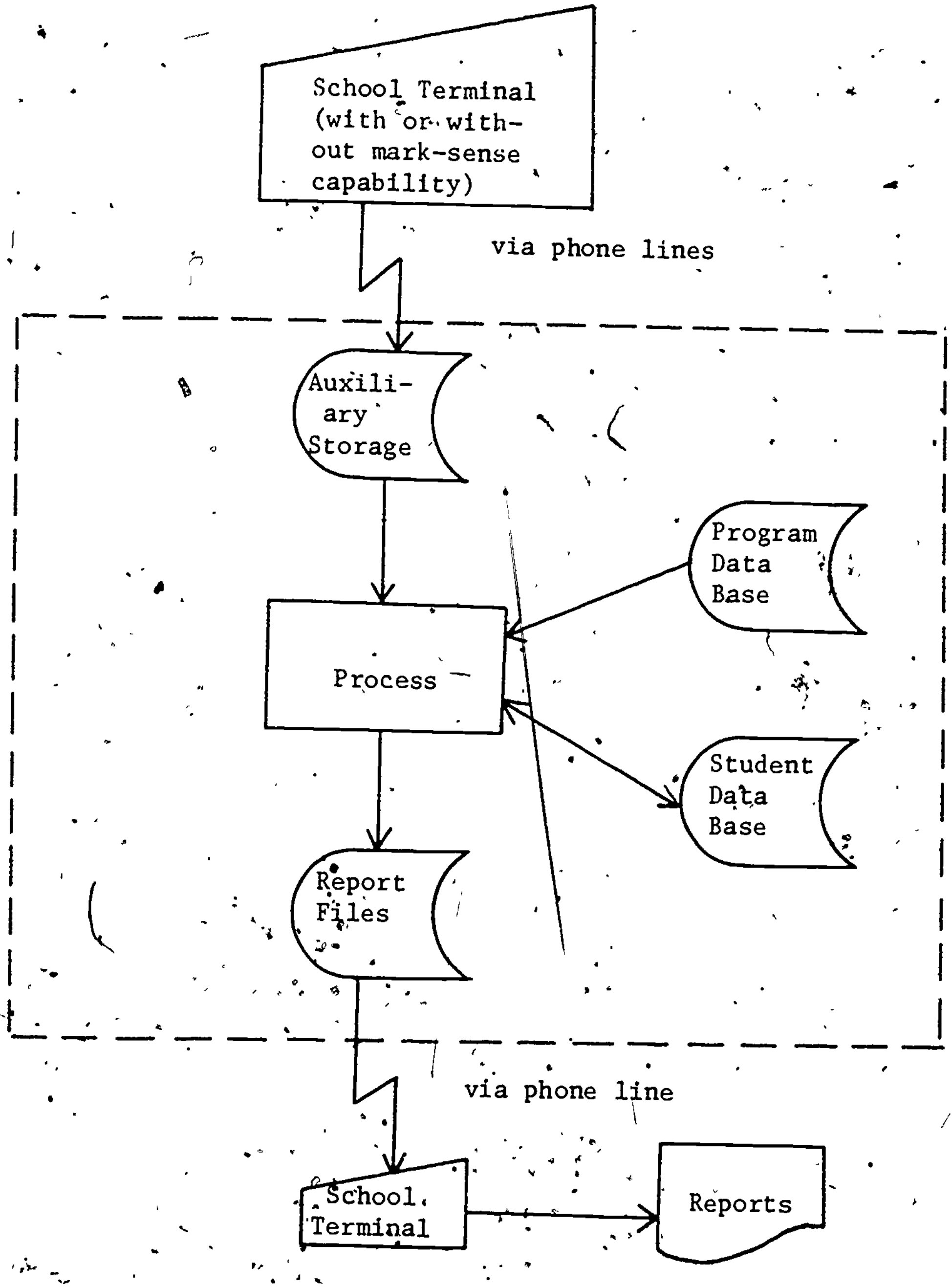


Figure 33. On-line system with two-way telecommunication.

Note: Dotted lines enclose central computer facility.

computer facility. The interactive "front-end" software is a significant element in interactive systems.

The interactive front-end has complete control over communication between the computer and the terminal operator in the interactive mode (see Appendix F for a summary of interactive terminal functions). It cues the operator in making appropriate keying functions and can provide immediate error detection. The interactive front-end also captures the data that will be subsequently processed to update the school's data base and/or trigger the generation of reports.

The most immediate response is achieved when the school's data base is loaded at the time of interaction, as depicted in Figure 34. This allows the interactive front-end to initiate report generation or data base update while the appropriate school's data are immediately available to the computer. Thus, the processing of data and the transmission of results to the school can occur immediately. However, this immediate response uses a large amount of computer resources and limits the number of users which can be ultimately served by the system. In systems that have an extremely limited mass storage capability, this highly responsive approach will be able to accommodate only a very small number of users.

An alternative approach in the interactive mode of operation is to separate the data collection from the actual data processing. Then the actual updating of files or the generation of reports can be scheduled to take advantage of the availability of scarce computer resources. With this approach small computer configurations can service more users, and larger installations can operate more efficiently. The school's terminal operator would still communicate with the computer through an interactive front-end but only for submitting report requests or data required to

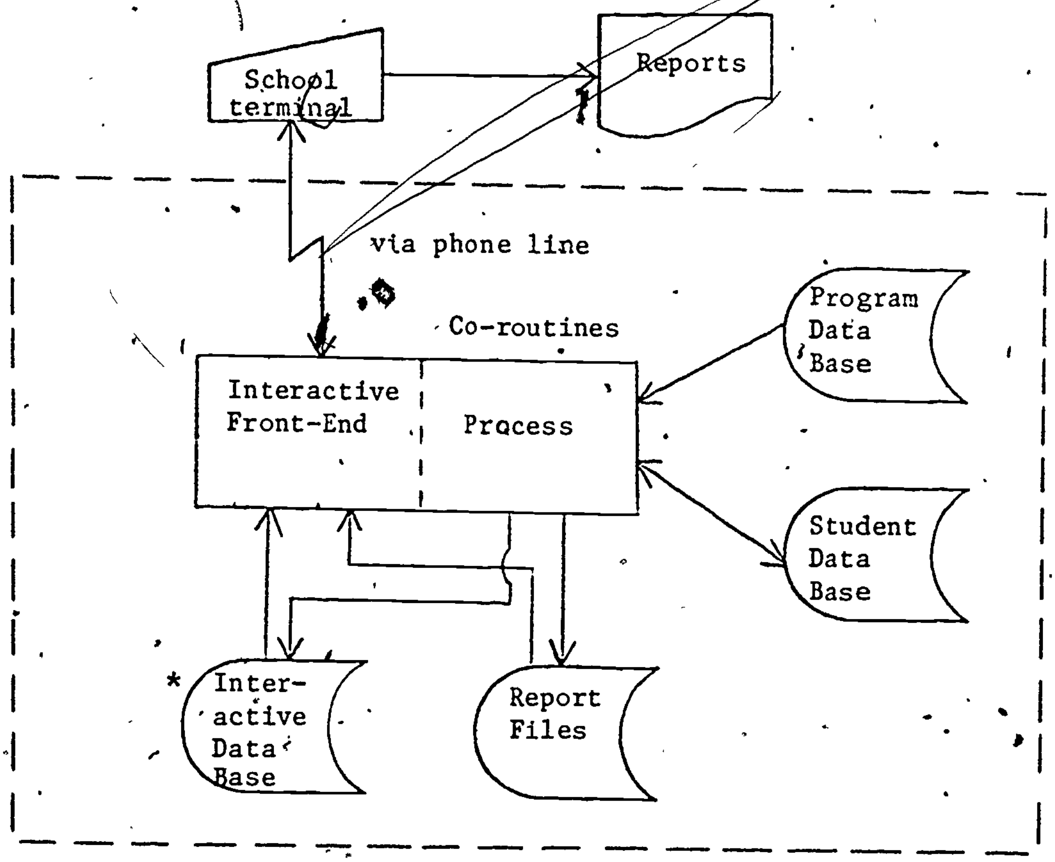


Figure 34. Interactive configuration with immediate processing of reports.

Note: Dotted lines enclose central computer facility.

*The /Interactive Data Base contains a small subset of the program data base and Student Data Base which includes school organization, student names and numbers, groups, and a skeleton curricula description. This data base enables error detection, error correction, and the generation of meaningful cues for the terminal operator.

update data bases. After all the data have been entered by the school's terminal operator, the interactive front-end would request the computer's supervisory program to schedule the CMI programs for processing when computer resources become available. The terminal is then disconnected from the computer. It is reconnected after the print files have been generated, at which time they are printed on the school's terminal.

The "delayed processing" approach appears to be most viable when computer resources are limited but when the advantages of interactive error checking are desired. Figure 35 shows an interactive configuration which permits report generating and file updating to be delayed until such time as appropriate computer resources become available. Since there is no requirement that the school's entire data base be loaded while a terminal is connected, the computer is able to support many users and perform other functions concurrently on a time-sharing basis.

WIS-SIM software is being developed in a modular manner in order to be compatible with the large range of computer configurations discussed in this chapter. This modular approach to software development will also expedite upgrading systems through the various operational modes.

This chapter has discussed design features of WIS-SIM which will enable it to be compatible with a wide spectrum of computer configurations. However, in order to be fully responsive to the design goal "to make a CMI system available to a large number of IGE schools," it would be helpful if an agency could be established to assist schools and service bureaus, when required, to adapt and implement the WIS-SIM design to their special requirements. The agency would act as an informational clearing house, provide consulting services, and sponsor workshops and seminars. Such an agency could take the form of a service bureau, a component of the various IGE

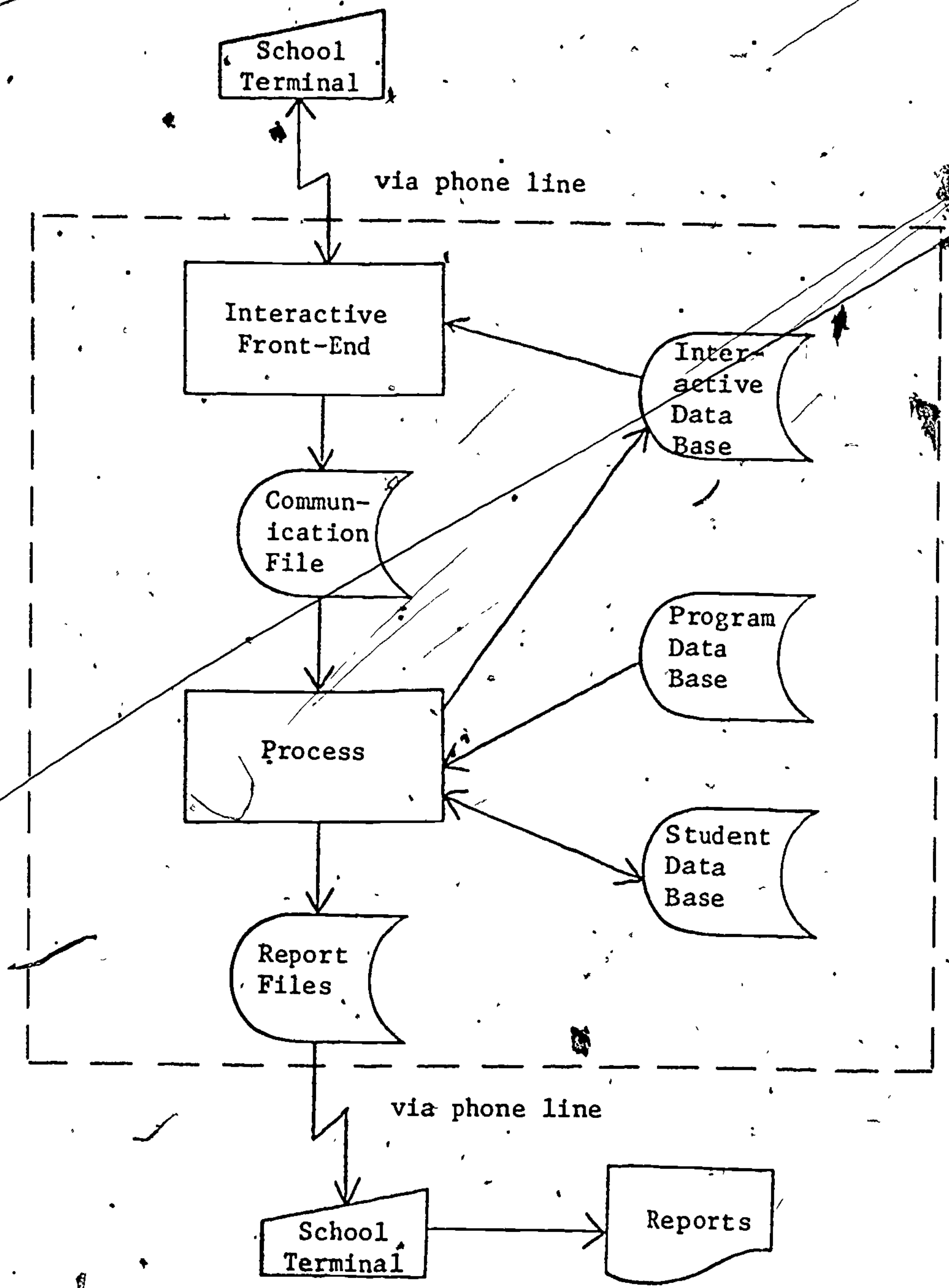


Figure 35. Interactive configuration with delayed processing of reports.

Note: Dotted lines enclose central computer facility.

Regional Centers, or a specialized National IGE Center could be established for this purpose.

VI

DEVELOPMENTAL SCHEDULE AND EVALUATION

DEVELOPMENTAL SCHEDULE FOR WIS-SIM

During the next several years, WIS-SIM will undergo further development, testing, and refinement. Because WIS-SIM is an integrated system of computer managed instruction, each individual product becomes more useful when it is integrated into the parent system. Therefore, it is necessary to develop, integrate, and test each addition to WIS-SIM through tryout, pilot, and field tests. In order to provide an overview of the specific developmental strategies planned, the major activities included in each phase will be outlined in the sections which follow. Specific dates cannot be affixed to each phase, since the current funding pattern is determined one year at a time.

Phase I

Phase I is not a projection of future activities; rather, it is a brief overview of what has been accomplished and what is presently taking place in the design of WIS-SIM. These activities include the development of two separate computer programs, with documentation and inservice training materials and procedures, for the management of the Wisconsin Design for Reading Skill Development and Developing Mathematical Processes (see Chapter I). These programs, and their attendant systems, are being tried out in two school districts. Additionally, WDRSD is being tested in a third school district. These programs have provided both a testing ground for formative evaluation of existing capabilities and information useful in the design and development of future enhancements. These enhancements include source data collection via mark-sense sheets, on-line and interactive capabilities, and further information reporting formats (see Chapter IV of this document and Computer Management of Individualized Instruction, Spuck, Hunter, Owen, & Belt, 1975).

The Wisconsin Research & Development Center has a great deal of flexibility in meeting its computing needs, including its own medium-scale computer (Harris 6024/5) and the large-scale computer (Univac 1110) of the Madison Academic Computing Center (MACC). Differential use of these computer facilities has been based on a policy which attempts to recognize the strengths of each computer system. The division of labor between the two systems is described in the Systems Plan-Increment IV Data Processing Equipment (1975).

Projected WIS-SIM activities, especially in the areas of research, administration (exclusive of classroom management) and instructional materials resource files require computational power, mass storage capability, and a sophisticated data base management system which are characteristic of large-scale computer systems such as the Univac 1110. It is necessary to interface the Harris Computer as an interpretive front-end to the Univac 1110 to ensure appropriate response and a high level of availability for the 16 on-line schools required for the pilot and field testing of the Generalized WIS-SIM. A high level of availability is especially critical during the development and evaluation of WIS-SIM. Thus, in Phase I a number of joint programming activities involving the R & D Center and MACC data analysis and programming staffs will take place. Systems programming will be required on both computers to implement the interactive front-end concept. Bench-mark programs of WIS-SIM applications will be developed and run on both computers in order to determine the best way to level the computing load of both machines. Appropriate modifications to the MACC Data Base Management System (DBMS) and the file organization of the WIS-SIM Generalized System will be made to optimize the compatibility between the two.

At the present time, MACC is making an extensive revision in its Data Base Management System. This revised Data Base Management System will have a variety of capabilities. It will be modular and incrementally upgradable. The system will handle hierarchical tree data base structures that can be accessed rapidly due to an extensive indexing capability. It will operate in both batch and interactive modes, has report generation and sequential file processing capabilities, and a checkpoint/restart feature to assure data integrity. Files may be accessed through an English-like query language or by user programs. Future developmental efforts in WIS-SIM will utilize this system for administrative reports and research activity.

Phase II

Emphasized in Phase II will be the design and development of the basic Generalized System, specifications for which have been presented in Chapters II and III. Developmental activities will result in the creation of Generalized System computer software, documentation, inservice materials, and a product evaluation plan. Performance profiling activities will include continued development of grade reporting methods and tryouts of these products. Development of grade reporting methods will be conducted in association with the R & D Center's Home-School-Community Relations Component and IGE Staff Development Project.

The diagnosing process, from the WIS-SIM model, will undergo extensive research leading to specification of algorithms useful for estimating student performance expectations. Performance expectations will be compared with actual performance to identify student instructional needs. Diagnosis will be aided by the production of "Management by Exception reports" to be utilized by teachers, administrators, parents, and students. These research efforts will result in a Technical Report outlining specific products, tactics, and areas for further research. Those diagnostic products now developed and documented will be pilot tested during this phase.

Research will continue on guiding the instructional program through two activities: a classification scheme for instructional activities and the monitoring and allocation of instructional activities. These activities will focus upon three areas: the integration of multiple curricula, school resource management, and instructional materials resource files. The latter area focuses on relating instructional activities to instructional objectives in order to assist teachers in selecting appropriate instructional activities and settings. The component's future activities in these areas will be delineated in a technical report.

Testing and test scoring will undergo feasibility assessment during this phase. These activities will lead to a technical report specifying the desirability of further development in these areas.

Developmental research and design on the evaluation, resource management, administration, and research components of WIS-SIM will also take place during Phase II. Concepts involved in the evaluation of curriculum content, prerequisite structures, instructional activities, and materials will be investigated. Administrative reports include summaries of student achievement across administrative levels larger than the classroom unit (school, district, etc.) and changes in student classification (new students, students transferring, etc.), providing essential, continuous monitoring and error checking. Design of the CMI research capability will focus on an ad hoc query capability. This will permit the relatively easy aggregation of information in the data base in ways which have not been formalized as regular reports. The research component will allow for an additional data base, containing data specific to the research which is to be undertaken, to be added temporarily to the system.

Phase III

The basic Generalized System which was begun in Phase II will be extended and tested further in Phase III. The software, forms, and inservice program for grade reporting activities of the performance profiling process will be pilot tested. These materials will be improved as formative evaluations suggest. All changes will be appropriately documented. The inservice program for grade reporting will include both school staff and parents.

Enhancements of the diagnosing component of the system identified in the technical report to be written under Phase II will be further refined; the software, documentation, and forms necessary to them will be developed. Documentation will continue to be written on three levels: user, technical, and system. An inservice program will be planned, developed, and presented to guide school personnel in the use of these enhancements. The tryout of these parts of WIS-SIM will begin in Phase III.

Activities related to the process of guiding the instructional program will be built upon both this report and the Theoretical Paper (Spuck, Hunter, Owen & Belt, 1975) written in Phase I. The requirements and the designs specified in those documents will lead to WIS-SIM enhancements in three primary areas: the integration of multiple curricula, school resource management, and instructional materials resource files. The developmental work in school resource management will be focused on program budgeting and will be carried out in cooperation with the staff of the Cost Effectiveness Component (R3) of the Wisconsin Research and Development Center. This research will provide costing and budgeting information concerning each of the instructional programs included in the school's program. The Phase III tryout tests will then include improved reports and services to assist teachers in selecting appropriate instructional activities and settings.

Program evaluation activities, research capabilities, and administrative reports will be undergoing developmental work. The addition of an ad hoc data base query capability to the management system will greatly enhance its research and evaluation potential. Research and evaluation activities, as defined in Phase II, include curricular programs, materials, and instructional evaluation activities; research into both CMI and programs of individualized education; and reports summarizing both student achievement and changes in student status. In Phase III, these activities will enter a developmental stage. Both system documentation and tryout will occur.

The Technical Report written during Phase II will guide the development of a test scoring capability. Development of software, forms, and an inservice program and documentation will be undertaken. These will be documented on three levels: user manuals, technical manuals, and system documentation. An inservice will be planned, developed, and presented to system users with appropriate materials. The tryout/pilot test of these products will be begun in Phase III.

Phase IV and Beyond

While additional refinements will continue to be made to the Generalized System, Phase IV will emphasize field testing of the system and its various components. Performance profiling will begin to be field tested and a final report on the various types of profiles, including the grade reporting format, will be completed. Software and its documentation will be revised to include improvements made apparent by the field test. Again, improvements in the content and format of information based upon on-going formative evaluation will continue, though end products will be available.

The diagnosing process will continue to be developed in the areas of identification of student needs at both the individual student level and at the school and district levels. The pilot test of system enhancements identified in Phase III will be completed and documented. The diagnosing component will become a part of the Generalized WIS-SIM field test.

The formative data, collected during the Phase III tryout, on the integration of multiple curricula, school resource management, and the creation of instructional materials files will lead to improvements in WIS-SIM capabilities in guiding the instructional program. These improvements will appear in the tryout report and will become part of the Generalized WIS-SIM pilot test.

The further development of WIS-SIM capabilities in program research depends in large part on when the <u>ad hoc</u> query capability is acquired. Little productive development can go beyond the theoretical design stages without this capability. Program evaluation and administrative reporting capabilities will be pilot tested, based upon the developmental activities in Phase III. This test will become part of the Generalized WIS-SIM pilot test and will be expanded to as many schools as feasible since these capabilities will be used generally at the option of the individual school.

On-line testing and test scoring will enter the pilot test stage during Phase IV. The report of the pilot test will guide formative evaluation and help integrate on-line testing into the complete Generalized System. In Phase IV, the reiterative process will continue and the products will be improved at each step. This approach includes need identification, preliminary product development, field testing, product improvement, pilot testing, formative evaluation, and further product refinement. While deliverable end items will be available in Phase IV, formative evaluation will allow constant improvement.

EVALUATION DESIGN FOR WIS-SIM

The primary goal of this project is to design, develop, implement, and evaluate a system of computer managed instruction supportive of programs of individualized education.

A second goal of the CMI project is to do developmental research on computer managed instruction, both after the development of the system and during the planning and design phases of evaluation. Thus, CMI research will often have a built-in evaluative focus. This focus will direct project activities through a needs assessment, preliminary product development, pilot testing, product improvement, field testing, and through the reiterative cycle; that is, decisions about product development are directly tied to the results of formative evaluation.

Two different decision areas necessitate two different kinds of evaluation. Formative evaluation is used to guide the ongoing development of the system. The relevant data are routinely collected during system operations. Summative evaluation is an overall assessment used to determine whether the goals of the program are being met. The broad objective of CMI is to collect and process student information to aid in making efficient and optimal decisions for each student in programs of Individually Guided

Education. If this objective is indeed being met, it is anticipated that it will be reflected through increased student achievement and student-teacher interaction. Since these are ideal outcomes that take place gradually, considerable time is required before they become directly assessable. Another reflection of the broad CMI objective would be decreased amount of teacher time spent with clerical work. This outcome is more easily subject to assessment and is discussed under formative evaluation.

Formative Evaluation

The formative evaluation design is subdivided into three sets of data (functioning, utilization, and effect) relevant to the goals of planning, developing, and implementing a system of computer managed instruction. Each set of data is further divided into three levels of consideration: actual, perceptual, and attitudinal. This reflects a desire to consider affective aspects of the program as well as the more objective effects of implementation (see Table 3).

Functioning and utilization evaluation provide information about how the program is being carried out and indicate changes or refinements needed in subsequent design and developmental stages.

Effect data consist of information provided through the ongoing operation of the system and include the various predictable outcomes from an implemented system of computer managed instruction.

Data collected at the "actual" level include hard information about the ongoing operation of the system. This would include turnaround time, numbers of system uses by type, and data on the effect of the system at the teacher level.

TABLE 3

EVALUATIVE CRITERIA MATRIX

	FUNCTIONING	UTILIZATION	EFFECT
ACTUAL	1. Turnaround time 2. New role descriptions	1. Number of system accesses 2. Usage of reports	1. Time spent in information management (decision making) 2. Knowledge of information content and use 3. Changes in cost per learning outcome ratio 4. Student learning outcomes will change positively.
PERCEPTUAL	1. Changes in school operations	1. Report format appropriateness A. Useful? B. Adequate?	1. Better decision making 2. Better time use 3. Better information for parents
ATTITUDINAL			1. Can WIS-SIM help to humanize the instructional process?

Data on the perceptual level are concerned with judgments about how the system is viewed. Feedback will be collected from administrators, teachers, and aides on questions such as the adequacy of inservice training, changes in staff roles, appropriateness of the reports provided, and responsiveness of the system to school needs.

The attitudinal level of consideration is designed to find out how school personnel and, potentially, parents and students, feel about CMI--which may be crucial to successful implementation. Attitudes of the staff toward computer involvement and the accompanying inservice training will be assessed, along with their confidence in dealing with CMI and the extent to which the system is meeting their expectations.

Formative evaluation assesses movement toward the goals of the program. It also facilitates the discovery of any positive or negative aspects of the program which were not anticipated in the original development. Progress evaluation corresponds to "Functioning" and "Utilization" in Table 3. As mentioned previously, the broad goals of the program as reflected in increased student achievement and student-teacher interaction are gradual changes and usually occur in small increments. Progress in the effect area is therefore designed to be evaluated on the basis of more immediate changes concerning how teachers are using the system. This should support speculation as to whether progress is being made toward the primary goals.

Data on the actual level of the effect area will be collected from teachers on a self-report questionnaire. Teachers will be questioned about how they use the forms provided with the CMI program (e.g., Do they make extensive additions and/or deletions to grouping reports, and if so, why? Do they use performance profiles for parent conferences, instructional planning, or other purposes?). They will also be questioned about time usage. How much of the teacher's time is occupied with planning, instructing, and clerical work?

Questions on the perceptual level will focus on teachers' judgments concerning the changes brought about by the CMI system. They will be asked whether the information provided is adequate and useful, whether they are using their time more effectively, and whether they see any changes in the quality of the student's educational experiences. Any other changes observed in school operations will be noted.

The third level of evaluative consideration in the effect area deals with the attitudes of the school staff toward the system. Have teachers accepted the computer reports or are they still using other grouping or record-keeping techniques? Do teachers feel free to make decisions on their own or do they feel bound by computer-generated suggestions? Has the computer system changed teachers' feelings of involvement in the instructional process or their relationships with the students?

Summative Evaluation

Summative evaluation will assess whether the system is providing better and more timely information to decision makers for optimizing each student's educational experiences. The primary goal of the system is to improve student achievement by making maximum use of the school's human and material resources and by implementing Individually Guided Education more completely. The system should not only provide useful information to teachers when they need it, but should also relieve them of some of the clerical burdens of record keeping. Teachers might have more time to spend with their students. This extra time could be spent in enrichment activities, motivational, tutoring, or goal setting procedures.

Evaluating these kinds of changes will draw on much of the data collected in the formative evaluation. For instance, the formative evaluation design can tell us whether teachers are indeed relieved from some time spent with clerical work, and whether they believe that the information provided by the system is useful in making instructional decisions. However, evaluating whether these changes are producing improved student achievement may be difficult.

Student achievement may be measured in various ways. Improved achievement may mean that students master more objectives in a given period of time. It may mean the objectives they master are learned more thoroughly, retained longer, or transferred better to related areas. Achievement may be assessed through program-specific measures or standardized measures. WIS-SIM will use program-specific measures, which will be a more sensitive evaluation of the system.

Since WIS-SIM, the CMI product being evaluated, is used to support individualization in specific instructional programs, evaluation designs should include control schools using the same instructional programs, but without a CMI system. The large number of intervening variables such as student variability, teacher variability, program implementation, and materials available make control school selection very difficult. Schools which are as similar as possible to the WIS-SIM schools will be included in the evaluation design for comparison purposes.

It might be possible to use data from previous years at the experimental schools for some comparisons. However, as teachers become familiar with these new and innovative instructional programs, they frequently revise the materials or change the way they use them. Further, new and different instructional materials are being used. This could make data from the beginning stages of implementation inaccurate or unreliable.

Summative evaluation is also concerned with what teachers do with the additional time a CMI system might provide. Do they use it in ways that are reflected in student improvement?

Finally, the design of the summative evaluation must consider "halo effects." Implementation of the system may result in increased attention to instructional programs or in other changes which are not directly related to any essential characteristics of the CMI system. In spite of the many difficulties, efforts toward designing a summative evaluation for WIS-SIM are continuing and include not only student achievement and instructional program aspects, but also focus upon cost-benefit and cost-effectiveness components. Only through such a design can the true value of WIS-SIM be determined.

CONCLUSION

This chapter has outlined four phases for current and future project activities. No specific dates have been attached to these phases but could represent time periods of approximately one year.

Although the project goal is to make a Generalized CMI System, with various enhancements and options, available to a large number of IGE schools, deliverable end products will be developed in each phase. Based on Phase I activities, computer programs and supporting materials for the two major IGE curriculum programs are currently available in batch, on-line, and interactive modes.

Finally, at the conclusion of each phase, a report will be written to review project activities and evaluations for the past phase and to revise and update those activities projected for ensuing phases.

REFERENCES

Belt, S. L., & Skubal, J. M. Individually Guided Education and the Wisconsin System for Instructional Management, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1974.

Belt, S. L., & Spuck, D. W. Computer applications in Individually Guided Education: A computer-based system for instructional management (WIS-SIM), Working Paper No. 125, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1974.

Evers, N. A., Karges, M. L., & Krupa, W. IGE Staff Development: Needs and Specifications, Working Paper No. 131, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1975.

Klausmeier, H. J. IGE: An Alternative Form of Schooling, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1974.

Klausmeier, H. J., Jeter, J. T., Quilling, M. R., Frayer, D. A., & Allen, P. S. Individually Guided Motivation, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1975.

Klausmeier, H. J., Quilling, M. R., Sorenson, J. J., Way, R. S., & Glasrud, G. R. Individually Guided Education and the multiunit elementary school: Guidelines for implementation, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1971.

Spuck, D. W., Hunter, S. N., Owen, S. P., & Belt, S. L. Computer management of individualized instruction, Theoretical Paper No. 55, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1975.

Systems Plan-Increment IV Data Processing Equipment, Madison: Wisconsin Research and Development Center for Cognitive Learning, 1975.

APPENDIX A

A Design for a Direct Access Curriculum Description File

APPENDIX A

A DESIGN FOR A DIRECT ACCESS CURRICULUM DESCRIPTION FILE

Two instructional programs (reading and math) are illustrated below. The reading program consists of two instructional modules (words and syntax) and the math program consists of two instructional modules (counting and units and measuring). The instructional objectives are listed under each module, (e.g., under the syntax module are listed three instructional objectives--nouns, verbs, and adjectives).

I. Reading

   A. Words

      1. Starting syllables

      2. Ending syllables

   B. Syntax

      1. Nouns

      2. Verbs

      3. Adjectives

II. Math

   A. Counting

      1. 1-10

      2. 11-20

   B. Units and measuring

      1. Small units

      2. Large units

      3. Measuring

The descriptions of these two instructional programs can be stored in a file as follows:

1	Reading
2	Math
3	Words
4	Syntax
5	Counting
6	Units & Measuring
7	Starting Syllables
8	Ending Syllables
9	Nouns
10	Verbs
11	Adjectives
12	1-10
13	11-20
14	Small Units
15	Large Units
16	Measuring

↑ Record Number

An intermediate mapping is generated to reflect the structure of the instructional programs and their modules/objectives.

MAP(1) = 2	Number of instructional programs
MAP(2) = 2	Number of modules in reading
MAP(3) = 2	Number of modules in math
MAP(4) = 2	Number of objectives in words
MAP(5) = 3	Number of objectives in syntax
MAP(6) = 2	Number of objectives in counting
MAP(7) = 3	Number of objectives in units and measuring

From this, a table of base addresses is built by performing cumulative additions.

ADDR(1) = 2

ADDR(2) = 4

ADDR(3) = 6

ADDR(4) = 8

ADDR(5) = 11

ADDR(6) = 13

for i = 1 to 6,

ADDR(i) = ADDR(i-1) + MAP(i)

where ADDR(0) = 0

This base address table can then be used to compute the record address of a descriptive record for any module or objective. By assigning numbers to the instructional programs, to the modules within programs, and to the objectives within modules, a record address may be computed as follows:

a. For instructional programs:

record address = program number

b. For modules:

record address = ADDR(prog. number) + module number

c. For objectives:

record address = ADDR(ADDR[prog. number] + module number) + objective number

For example, the record address of the objective small units

= ADDR (ADDR[2] + 2) + 1

= ADDR (4 + 2) + 1

= ADDR (6) + 1

= 13 + 1

= 14

APPENDIX B

Prerequisite Coding

APPENDIX B

PREREQUISITE CODING

Prerequisites must be coded into machine readable form before a computer program can use them to make instructional grouping recommendations. There are several possible approaches to coding these prerequisites. The approaches include a boolean algebra structure of AND and OR conditions, a partial ordering structure, and a list of parameters for a prewritten scanning algorithm. From the standpoint of generality and coding simplicity, the last approach is most desirable and only it will be described below.

Typically, prerequisite behavior is recommended before instruction is begun on any module (objective) within a given instructional area. This recommended behavior should be translated into a set (or sets) of prerequisite conditions based on prior assessments. To illustrate this, a two-level scanning algorithm of prerequisite conditions will be described followed by a description of how recommended behavioral assessments are coded into parameters for this algorithm.

The first step of the algorithm is to check the current assessment(s) of the objective (module) for which the grouping has been requested. If a student has mastered this objective (module), he is not included in the recommended grouping. If a student has not completely mastered this objective (module), his prior assessments will be scanned to determine if he "passes" the prerequisites. If he does pass the prerequisites, he will be included in the recommended grouping.

In order to pass the prerequisites, a student must meet a specified number of conditions. The number of conditions a student must meet is given by two parameters: (1) CMAX, the number of conditions he may be tested for, and (2) CMIN, the minimum number of these conditions he must satisfy in order to pass.

For each condition, there are six parameters which describe what scores are to be scanned, the minimum level of achievement, and the number of these scores which must be at least as good as the minimum level before a student is considered to have met this condition. These parameters are:

AREA--instructional area or program

MODULE--instructional module

OBJEC--first objective to be scanned

OMAX--number of objectives to be scanned

OMIN--number of objectives that must have scores better or equal to N

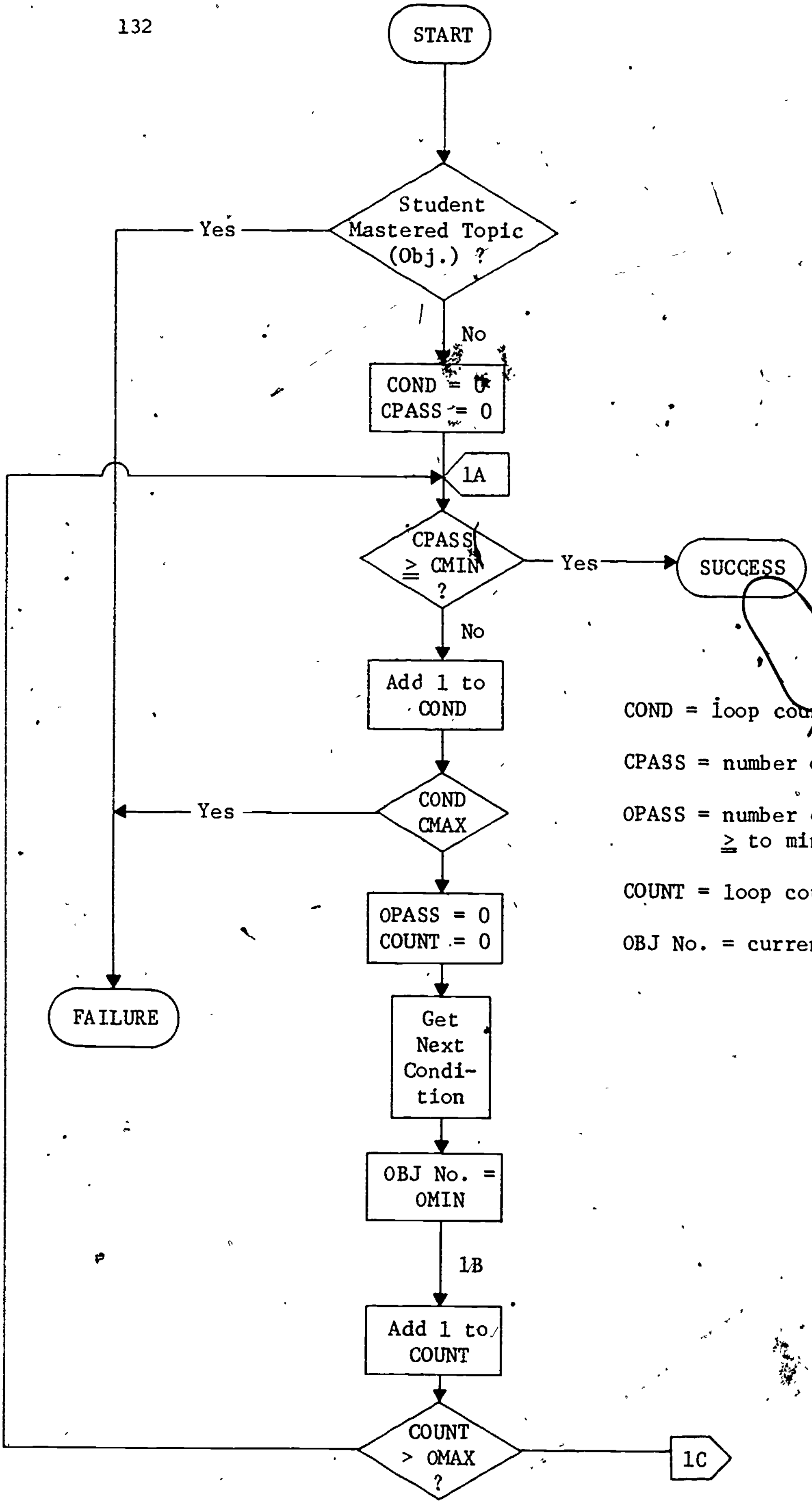
MSCOR--the minimum score

Figure B-1 presents a flow chart of this algorithm.

Once the scanning algorithm has been finalized and its parameters determined, the prerequisites must be coded from the recommended behavioral assessments. This can best be described through an example. Consider the prerequisite behavior for topic 28 in DMP:

> "The prerequisite for this topic is mastery or progress toward mastery of the objectives in topic 20. A child should also have experienced some of the activities of topic 26."

This statement specifies two conditions, both of which must be met before a student can be considered as "passing" the prerequisite. Therefore, both parameters, CMAX and CMIN, will be coded with the value 2.



COND = loop counter for conditions

CPASS = number of conditions met

OPASS = number of objective scores ≥ to minimum

COUNT = loop counter for objectives

OBJ No. = current objective

(continued on next page)

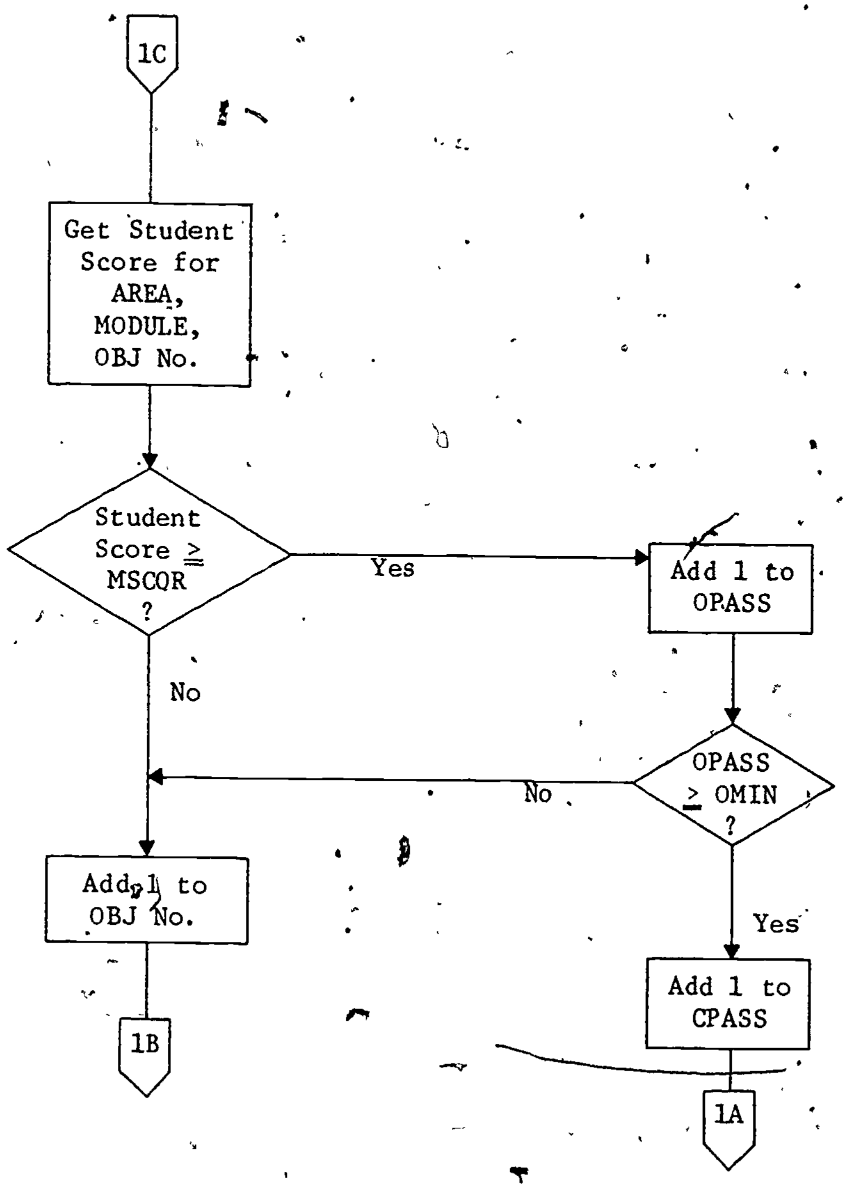


Figure B-1. Flow chart of algorithm for checking prerequisite structure (continued).

The first condition specifies "mastery or progress toward mastery of the objectives in topic 20." In DMP, scores are represented at four levels:

M = Mastery (or TC = Teacher Certification)

P = Progress toward mastery

N = Needs help

NA = Not assessed

Since there are 3 objectives for topic 20, the six parameters needed to specify this condition would be:

AREA = the number designated for DMP

MODULE = 20

OBJEC = 1

OMAX = 3

OMIN = 3

MSCOR = 'P'

Note that OMAX = OMIN = 3; this means that all the objectives of the topic must have a grade of 'P' or higher.

The second condition specifies that "a child should also have experienced some of the activities in topic 26." The word "experienced" here is assumed to mean that the student has had some instruction which would be indicated by any assessment (i.e., 'N' or higher). Since there are only two objectives in topic 26, an assessment of either of them would satisfy the requirement for "some of the activities." Therefore, the six parameters needed to specify this condition would be:

AREA = the number designated for DMP

MODULE = 26

OBJEC = 1

OMAX = 2

OMIN = 1

MSCOR = 'N'

To summarize this particular prerequisite, the coding would be as follows:

CMAX = 2			CMIN = 2			
1st Condition	DMP Number	20	1	3	3	'P'
2nd Condition	DMP Number	26	1	2	1	'N'

To more accurately reflect the coding of the prerequisite, the prerequisite description may be recomposed as

"M or P rating on objectives 1-3 of DMP topic 20 and assessment on at least one of objectives 1 and 2 of DMP topic 26."

Any remarks which may be useful to the instructor for determining student readiness may be added to the prerequisite description.

APPENDIX C

Coding of Objective Overlap (IOE File)

APPENDIX C

CODING OF OBJECTIVE OVERLAP (IOE FILE)

Each Instructional Objective Equivalency (IOE) file consists of items of equivalent relationships. The KEY to each item is the instructional module number and the number of the objective that is overlapped by instructional objectives in other instructional programs. These items are arranged sequentially in ascending order of the keys. Therefore, whenever the equivalent instructional objectives of a given instructional objective are needed, the appropriate IOE file can be searched sequentially by using the module number and the objective number.

Overlap can occur in a number of ways. Using DMP and SAPA overlaps as examples, three types of equivalency relations are found:

1. One-to-One Equivalency Relation. For example, DMP Topic 1, Objective 1 = SAPA Module 3, Objective 1.

   Whenever a student scores mastery in DMP Topic 1, Objective 1, mastery notation (M) is made in the DMP record of Topic 1, Objective 1 and equivalent mastery notation (MX) is made in the SAPA record of Module 3, Objective 1, and vice versa.

2. Multiple-to-One Equivalency Relation. For example, DMP Topic 2, Objectives 1 and 2 = SAPA Module 8, Objective 1.

   Whenever a student scores mastery in DMP Topic 2, Objective 1, mastery notation (M) is made on the DMP record only, unless the student has also scored mastery (M) in DMP Topic 2, Objective 2, in which case equivalent mastery notation (MX) will be made on the SAPA record of Module 8, Objective 1. DMP Topic 2, Objective 1 is a complementary instructional objective to Topic 2, Objective 2 and vice versa.

3. One-to-Multiple Equivalency Relation. For example, DMP Topic 22, Objective 1 = SAPA Module 23, Objectives 1 and 2.

   Whenever a student scores mastery in DMP Topic 22, Objective 1, mastery notation (M) is made on the corresponding DMP record and equivalent mastery notations (MX) are made on two SAPA records, Module 23, Objective 1 and Module 23, Objective 2.

From these examples, it is obvious that equivalency relation may be M-to-N, where M and N are any positive integers. Furthermore, since the Generalized System is designed to manage any number of Instructional Programs, an equivalency relation may be M-to-N-to...,-L, where M, N,...,L are any positive integers. An item, then, is the coding of any such equivalency relationships in an Instructional Program.

The components of each item are:

i. KEY to this item.

ii. number of complementary instructional objectives, NCOMIO.

iii. complementary instructional objective descriptor, COMDES.

iv. number of equivalency descriptors, NEQDES.

v. equivalency descriptor, EQDES.

where

KEY is a pair of numbers (n, m) such that n and m are the module number and objective number of an instructional objective that is overlapped by instructional objectives from other instructional programs.

NCOMIO is an integer which represents the number of complementary instructional objectives that the instructional objective, identified by the KEY, possesses.

COMDES is a pair of numbers (i, j) representing the module number and objective number of a complementary instructional objective of the KEY.

NEQDES is an integer which represents the number of overlaps that the instructional objective, as described by the KEY and the COMDES descriptor of this item, has. Therefore, for an M-to-N-to-L equivalency relation, NEQDES would be 2.

EQDES is a two-level equivalency descriptor to identify the combination of instructional objectives of an instructional program that overlaps the instructional objectives as described by the KEY and COMDES descriptors of this item.

Level 1 consists of (1) the Instructional Program Code, IPCODE, and (2) the number of instructional objectives of this program that form the combination, NEQIO.

Level 2 consists of as many equivalent instructional objective descriptors, EQDES, as the content of NEQIO. Each EQDES is a pair of numbers representing an instructional module number and objective number.

The following example illustrates how an IOE file can be composed: Assume that the Generalized System manages the WDRSD, DMP, and SAPA programs, and that the instructional program codes are:

Study Skills (SS) = 1

Word Attack (WA) = 2

Comprehension (COMP) = 3

DMP = 4

SAPA = 5

Following is the table of equivalent instructional objectives:

SS	WA	COMP	DMP	SAPA
Level C, Skill 5			Topic 5, Obj. 2	Module 6, Obj. 3
			Topic 7 Obj. 1 & 2	Module 9, Obj. 3 Module 10, Obj. 1
		Level B, Skill 1	Topic 8, Obj. 4	Module 8, Obj. 1 and Obj. 3

From this table, a table of equivalency relations can be constructed for each of the instructional programs. These equivalency relations can then be coded into items of the corresponding IOE files.

Following is the table of equivalency relations for the DMP Program:

Topic 5, Obj. 2	SS Level C, Skill 5	SAPA Module 6, Obj. 3
Topic 7, Obj. 1 & Obj. 2	SAPA Module 9, Obj. 3 & Module 10, Obj. 1	
Topic 8, Obj. 4	COMP Level B, Skill 1	SAPA Module 8, Obj. 1 & Obj. 3

From this table, an IOE file for DMP can be coded. Relation 1 is coded into item 1 with

KEY = (5,2)

NCOMIO = 0

No COMDES

NEQDES = 2

EQDES = (1, 1, (3, 5)) and (5, 1, (6, 3))

Relation 2 is coded into items 2 and 3, with

Item 2: KEY = (7, 1)

NCOMIO = 1

COMDES = (7,2)

NEQDES = 1

EQDES = (5, 2, (9, 3), (10, 1))

Item 3: KEY = (7, 2)

NCOMIO = 1

COMDES = (7, 1)

NEQDES = 1

EQDES = (5, 2, (9, 3), (10, 1))

Relation 3 is coded into item 4 with

KEY = (8, 4)

NCOMIO = 0

No COMDES

NEQDES = 2

EQDES = (3, 1, (2, 1)) and (5, 2, (8, 1), (8, 3))

The IOE files will be made up of records of fixed length, each containing an item, by storing the contents of KEY, NCOMIO, COMDES's, NEQDES, EQDES's in consecutive words of the record. Thus the IOE file of DMP has the following structure:

Record 1	5	2	0	2	1	1	3	5	5	1	6	3				
Record 2	7	1	1	7	2	1	5	2	9	3	10	1				
Record 3	7	2	1	7	1	1	5	2	9	3	10	1				
Record 4	8	4	0	2	3	1	2	1	5	2	8	1	8	2		
Record 5	0															

Record 5 contains 0 in the first word to signify the end of this IOE file.

From the example above, it should be clear that IOE files can be tailored to fit any local adaptations as described in Chapter II.

APPENDIX D

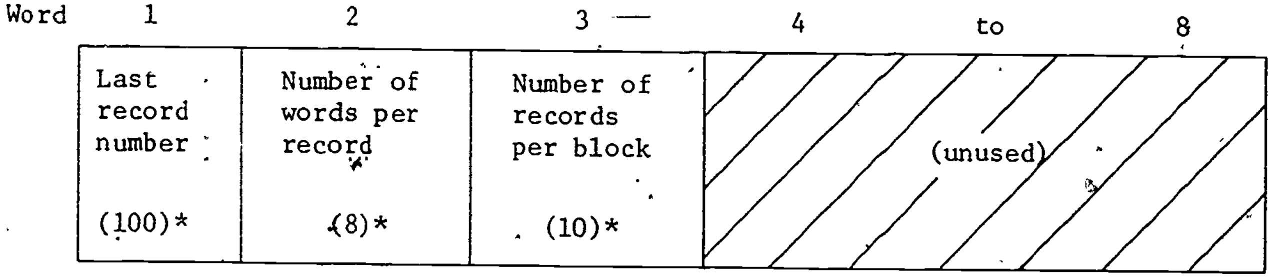
Unit-Teacher File

APPENDIX D

UNIT-TEACHER FILE

The Unit-Teacher file contains the sizes of units A to J (expressed in number of students), and the names of teachers (numbered from 1 to 9) assigned to each unit of the school. The size of a unit is used for two purposes: (1) to determine whether a unit contains any students; and (2) to estimate the file sizes needed to hold reports of the unit. Hence, for an empty unit, the size would be 0, for a non-empty unit, the size should be approximate but larger than the actual number of students, so that deleting or adding a few students from/to a unit in the middle of a school year will not necessitate an update to the file. A teacher's name has to be retrieved when only his or her unit code and number are given and his or her name is needed, as in the case of generating Instructional Grouping Inserts.

This file is organized as a random access file with records that are 8 words long and numbered from 0 to 100. Record 0 is used as a file-header, which contains the last record number, the number of words per record, and the number of records per block in the first three words of the record as shown in Figure D-1. Records 1 to 100 are divided into 10 blocks, each consisting of 10 consecutive records containing data pertaining to a unit. The block in which data of a particular unit are stored is determined by the order of its unit code in the alphabet (e.g., the data of unit A is stored in the first block and that of unit C in the third block, as illustrated in Figure D-2). It is possible that some units may not have any students and, therefore, no teachers would be assigned to them; a block



Word	1	2	3	4 to 8
	Last record number	Number of words per record	Number of records per block	(unused)
	(100)*	(8)*	(10)*	

*The numbers in parentheses are the actual numbers stored.

Figure D-1. File-header format.

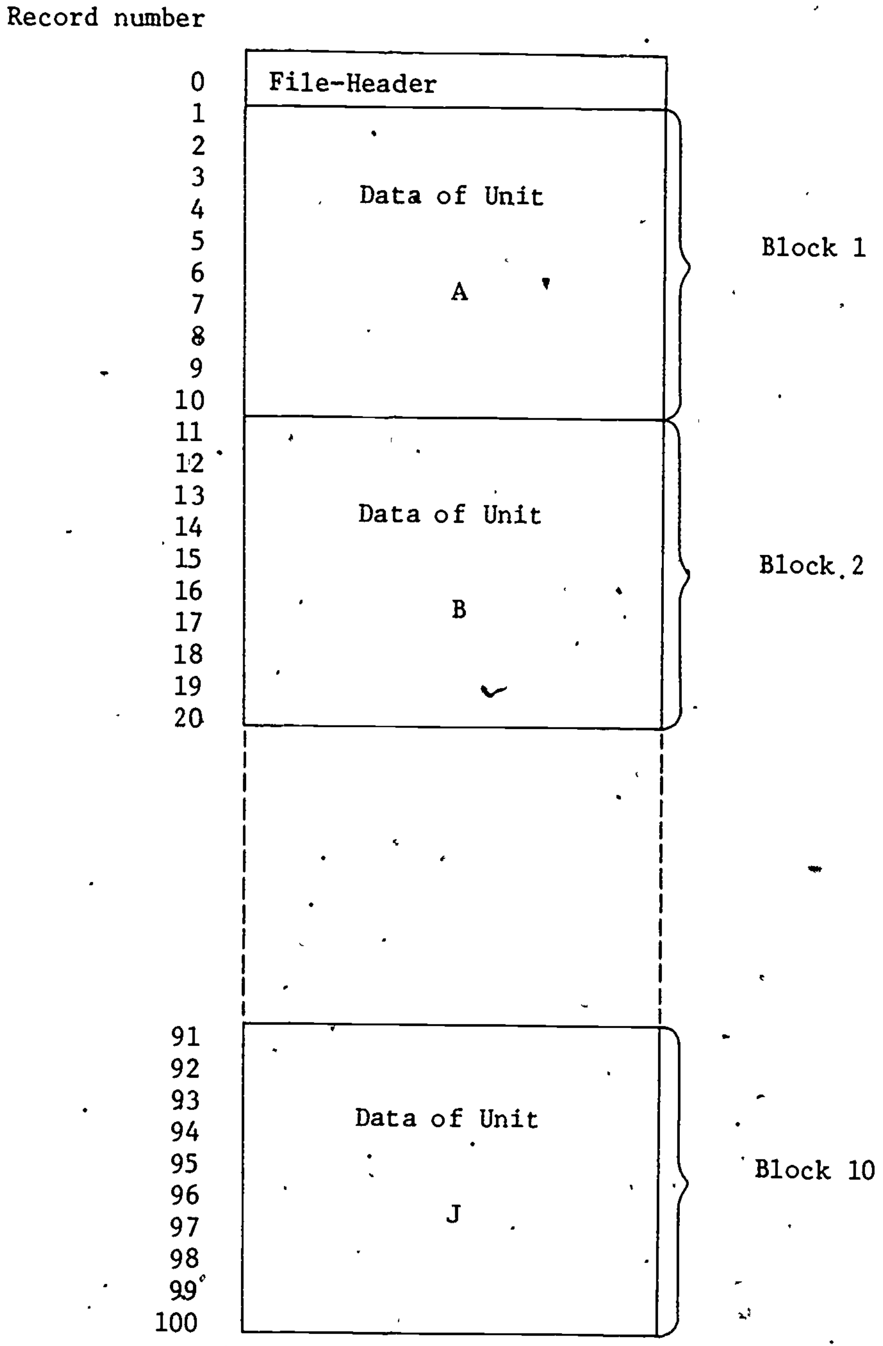


Figure D-2. Blocked structure of the Unit-Teacher file and sequencing of unit data in the file.

assigned to such a unit would contain a 0 as the first word of all its records. A block assigned to a non-empty unit has the unit size in the first record and the name of the teacher with number N in the (N+1)th record of the block, as illustrated in Figure D-3. Since nine or fewer teachers may be assigned to a unit, the record for an unassigned teacher number in that unit's block would be 0 for its first word; the rest of the record would be unused.

For a school with fewer than ten units and/or fewer than nine teachers assigned to each non-empty unit, many of the records between 1 to 100 will be unused; this waste of space is justified because in most computer configurations the minimum disc area size that must be assigned to a single file is much larger than the 808 words or 2424 bytes occupied by this file. The form described for the Unit-Teacher file enables direct access to the size of a unit or the name of a teacher through the number of the record that contains the data. The record number (RNO) for a teacher can be computed when his or her unit code and teacher number are given by using the following equation:

$$RNO = (MAPUNT(UNIT) - 1) * 10 + TEANO + 1$$

where MAPUNT(UNIT) is a function that converts the given unit code (A to J) to its order in the alphabet (1-10) and TEANO is the given teacher number. The number of the record containing the size of a given unit can also be computed with the above equation; TEANO is assumed to be 0.

	1	2	3	4	5	6	7	8
1		(unused)						
2	Name of teacher numbered 1 (left justified)							
3	Name of teacher numbered 2 (left justified)							
4	Name of teacher numbered 3 (left justified)							
5	0	(unused						
6	Name of teacher numbered 5 (left justified)							
7	0	(unused)						
8	0	(unused)						
9	0	(unused)						
10	0	(unused)						

Figure D-3. Data format of a block assigned to a unit having teachers numbered 1, 2, 3, and 5.

APPENDIX E

Student Data Base File

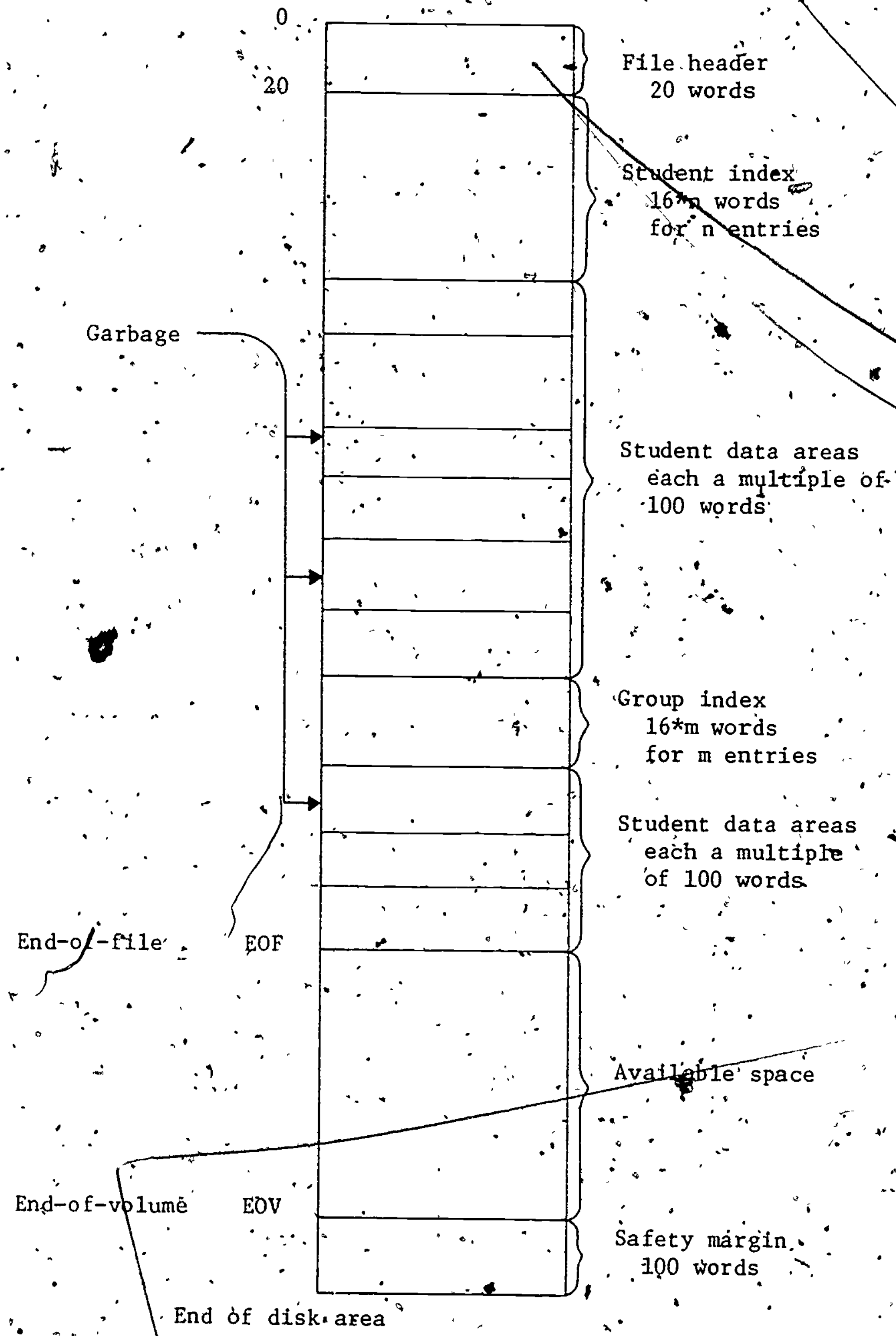
APPENDIX E

STUDENT DATA BASE FILE

To accommodate the variable length and content of student performance and demographic data under the WIS-SIM Generalized System, the Student Data Base (SDB) file is organized by the Unified File Structure (UFS) so that the existing UFS Management System, developed at the University of Wisconsin Research and Development Center, can be used to access the Student Data Base. This SDB file is organized as a direct access file and consists of four components: the file header, the student index, the group index, and the student data areas. Figure E-1 shows the layout of the SDB.

The file header is a block of 20 words at the beginning of the file. It contains the file identity, the dimensions of the file, and the indices. The contents of the file header are shown in Figure E-2.

Each student and group index is composed of 16-word index entries residing linearly in the index as shown in Figure E-1. Each index entry has an associated index number which is the order of the index entry in the index. The first eight words of each index entry make up the key field. To facilitate and speed searching through the index entries of each index, a binary tree structure, ordered by the key, is built in (by storing the index numbers of the parent, predecessor, and successor index entries in the last three words of each index entry). The binary tree is ordered so that for each index entry, the number of the key of its predecessor is less than the number of the key of the entry, and the number of the key of its successor is greater than the number of the key of the entry. Figure E-3

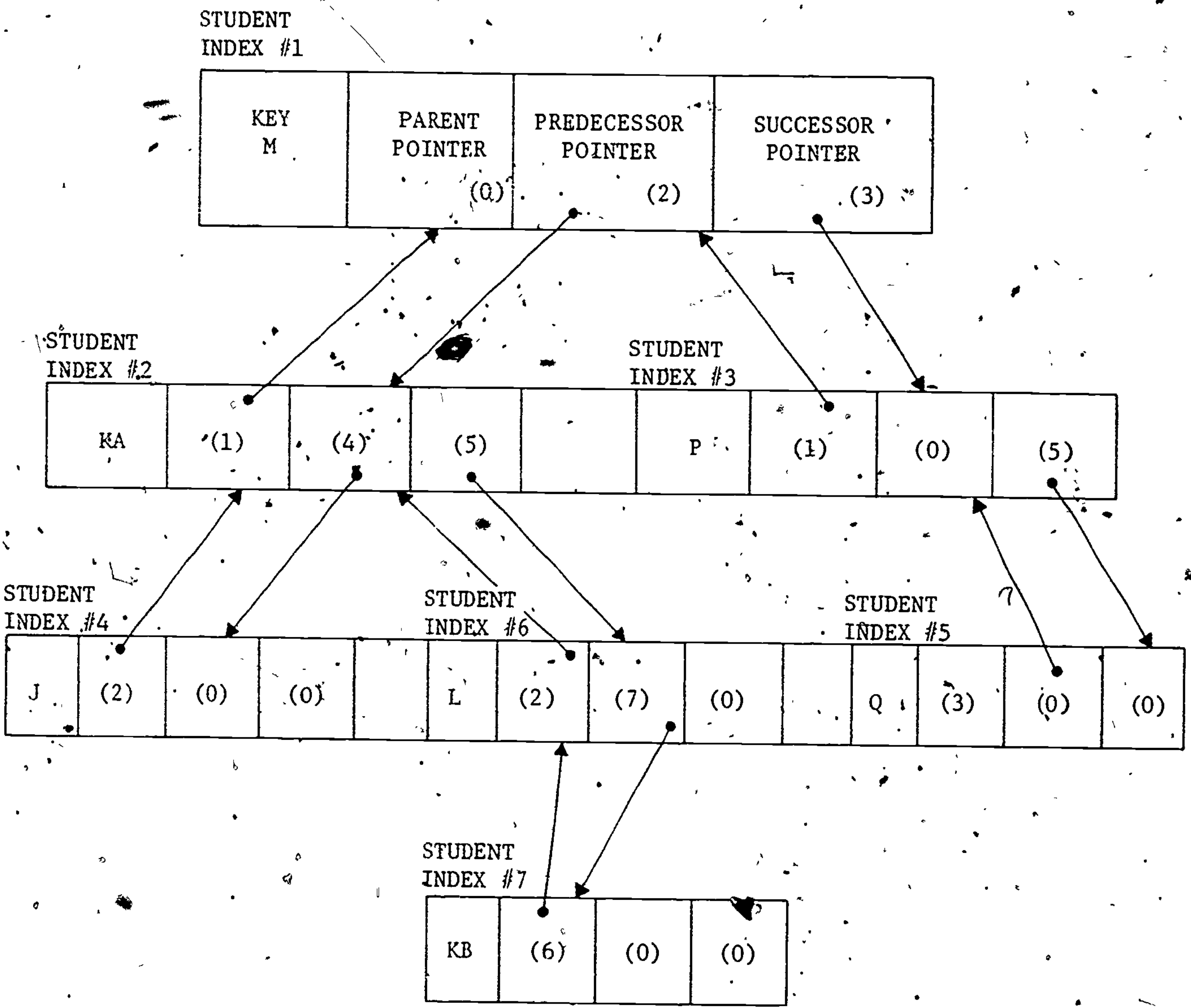


*The FORTRAN convention for multiplication.

Figure E-1. SDB file layout.

Word		
1	Date Last	
2	Packed	
3	Version number	
4	End of file address	EOF
5	Number of garbage words	
6	Size of	
7	Number of entries in	STUDENT INDEX
8	Start address of	
9	Avail chain pointer for	
10	Root index number of	
11	Size of	GROUP INDEX
12	Number of entries in	
13	Start address of	
14	Avail chain pointer for	
15	Root index number of	
16	End of volume address	EOV
17	File identification	
18		
19		
20		

Figure E-2. Content of Student Data Base file header.



Each box represents an index entry in a Student Index consisting of seven entries. The index number implicitly associated with each entry appears above each box. The divisions in a box from left to right represent the key, parent pointer, predecessor pointer, and successor pointer fields of the entry. The letter in the key field of an index entry is the key; the numbers enclosed in parentheses are the parent, predecessor, and successor pointers which are the index numbers of the entries. If the number in any pointer field is 0, then there is no parent, predecessor, or successor entry represented by that box. Arrows illustrate the binary tree structure. Notice that the index entries reside linearly in the Student Index in the order of their index number, but if the index is traversed in post order (symmetric order) the index entries are linked in the alphabetical order of their keys.

Figure E-3. An example of the binary tree structure of the student index.

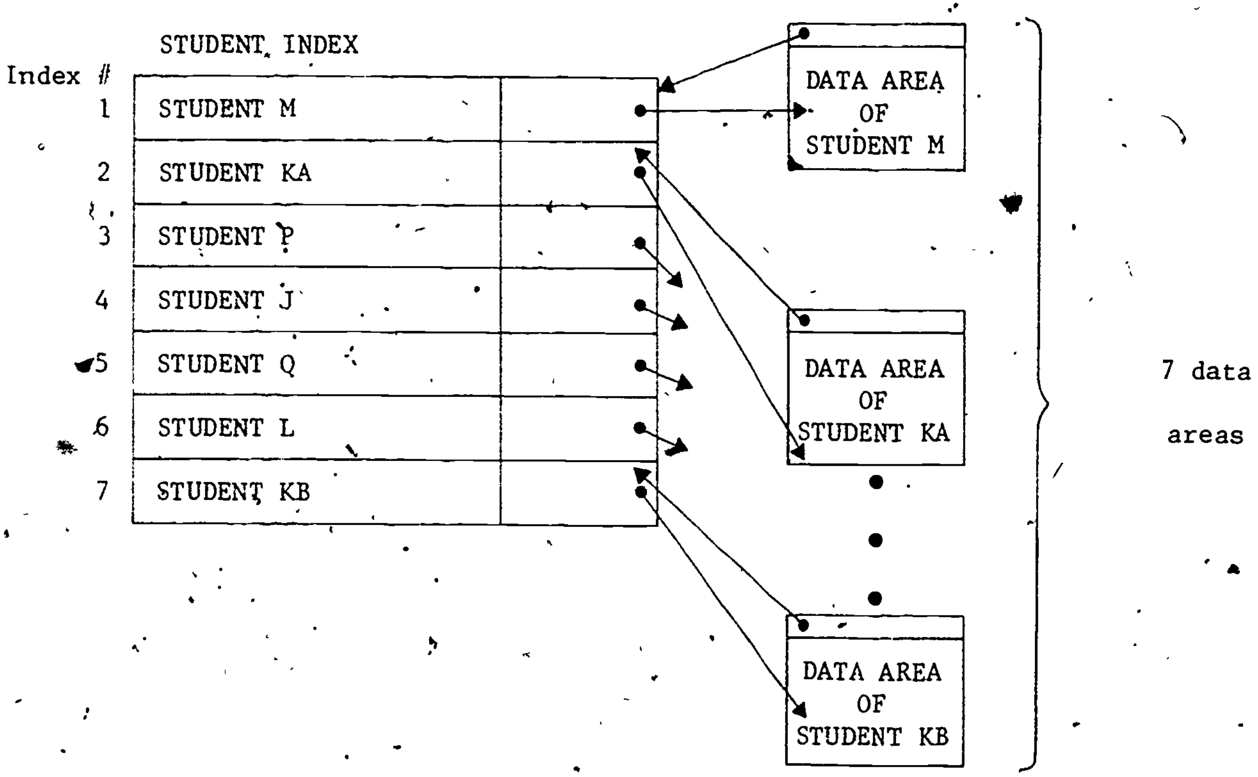
illustrates the binary tree structure of the student index; the group index has a similar structure.

Associated with each student index entry is a student data area containing all data records of the student whose name and unit are combined to form the key in the index entry. This association is expressed by storing the address and size of the student data area in the twelfth and thirteenth words of the index entry, and the index number of the entry in the first word of the student data area. The relationship between the student index and the student data areas is illustrated in Figure E-4. The format of each index entry in the student index is specified in Table E-1. Each group index entry designates a group of students whose group name is stored in the key field of the index entry. Members of a group are threaded into a doubly linked ring structure. This ring structure is implemented by storing the index numbers of the first and last students in the twelfth and thirteenth words of the group's index entry; in addition, for each student in the group, there is a group link record stored in the student's data area which contains the index numbers of the student before and the student after that student in the group. Table E-2 specifies the format for a group index entry for an instructional group, and Figure E-5 illustrates the ring structure associated with a group index entry and the relationship between the group index and student index.

The student data area can contain four different kinds of records: current, history, group link and simple. The first word in each of these records is the record key which consists of three fields:

KIND	AREA	TYPE

The KIND field is 1 for current records, 2 for history records, 3 for



The Student Index is composed of seven entries. The arrow from an index entry to its data area points to the address of the data area which is stored in a field of the entry. The arrow from a data area to its index entry points to the index number of the entry which is stored in the first word of the data area.

Figure E-4. An example of the relationship between the student index and the data areas of the SDB file.

TABLE E-1

FORMAT OF A STUDENT INDEX ENTRY

Data Item	Size	Content/Remark
Key	8 words	Char. 1-23 = student name Char. 24 = unit code
Scratch area	1 word	Usually used for storing flags when performing functions of WIS-SIM. Contains 0 when unused.
Sex	1 char.	F or M
Teacher	1 char.	Homeroom teacher number
Grade	1 char.	1-12
Birthdate	1 word	Char. 1 = month 2 = day 3 = last 2 digits of year
Pointers	5 words	Word number 1 = address of data area number 2 = size of data area number 3 = index number of predecessor index entry number 4 = index number of successor index entry number 5 = index number of parent index entry

TABLE E-2

FORMAT OF A GROUP INDEX ENTRY FOR AN INSTRUCTIONAL GROUP

Data Item	Size	Content/Remark
Group key	8 words	Word number 1 = unit code (left justified) number 2 = instructional program number number 3 = MAP (LEVEL, SKILL)/Topic number/Module number number 4 = Teacher number number 5 = Cycle number number 6-8 = 0 where MAP is a linear transformation of all pairs of (LEVEL,SKILL) in WDRSD. NOTE: This form of group key is for instructional groups only.
Scratch area	2 words	May be used to contain demographic data of the group if desired. Contains 0 when unused.
Number of students in group	1 word	
Pointers	5 words	Word number 1 = index number of last student in group number 2 = index number of first student in group number 3 = index number of predecessor index entry number 4 = index number of successor index entry number 5 = index number of parent index entry

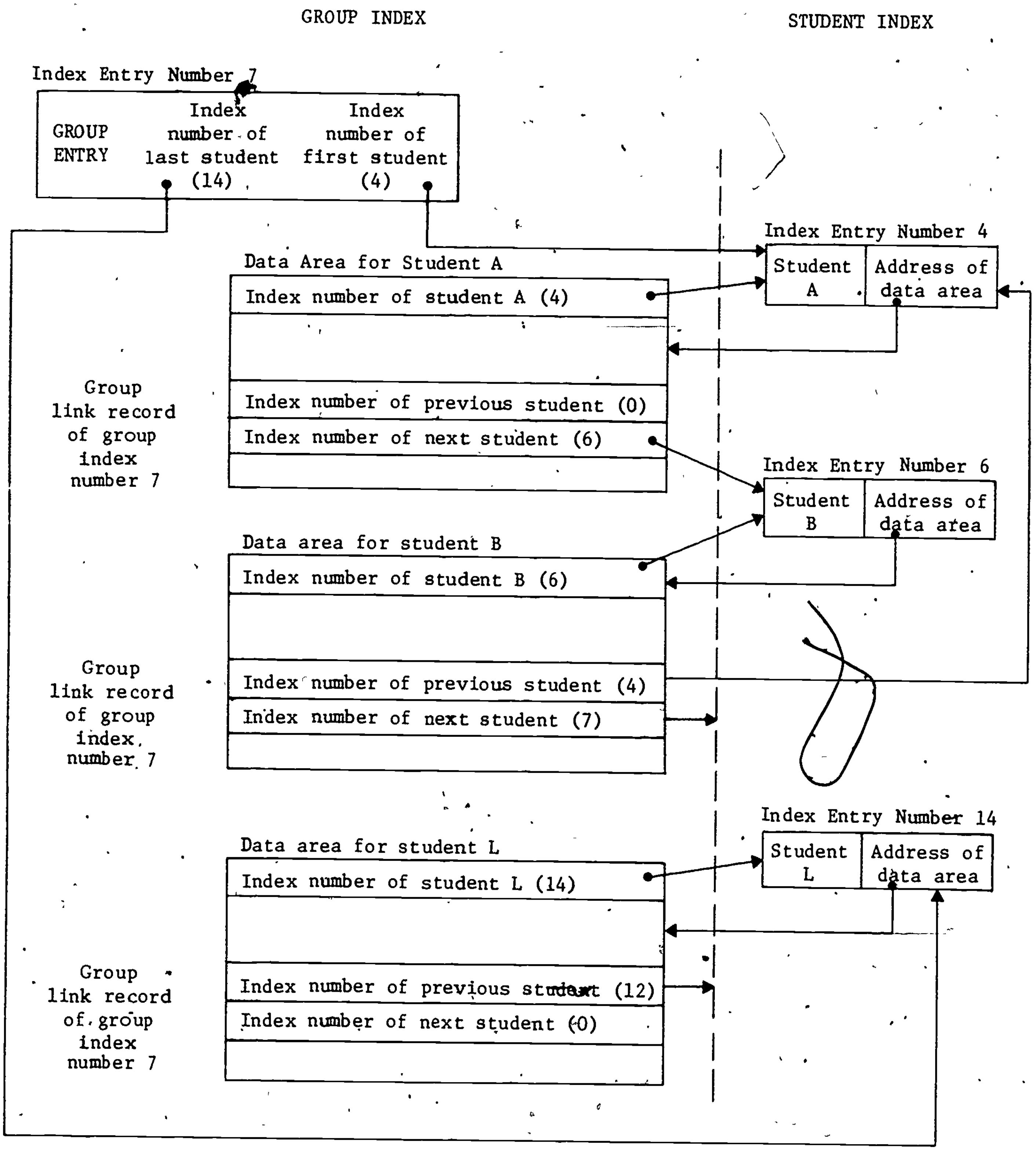

GROUP INDEX
STUDENT INDEX
Index Entry Number 7
GROUP ENTRY
Index number of last student (14)
Index number of first student (4)
Index Entry Number 4
Student A
Address of data area
Data Area for Student A
Index number of student A (4)
Index number of previous student (0)
Index number of next student (6)
Group link record of group index number 7
Index Entry Number 6
Student B
Address of data area
Data area for student B
Index number of student B (6)
Index number of previous student (4)
Index number of next student (7)
Group link record of group index number 7
Index Entry Number 14
Student L
Address of data area
Data area for student L
Index number of student L (14)
Index number of previous student (12)
Index number of next student (0)
Group link record of group index number 7

Figure E-5. Ring structure associated with a group index entry and the relationship between the group and student indices.

group link records, and 4 for simple records. For each KIND of record, there may be several records with different AREA or TYPE fields. For group link records, the AREA and TYPE fields are concatenated into one field which contains the group index number with which it is associated. The second word of each record contains its length (with the exception of group link records, whose length is always two words). For current and history records, the length is in words; for simple records, it is in characters. Each student data area can contain an unlimited number of records of each KIND in any order. The only requirement is that the key for each record be unique. Each student data area is allocated 100 words initially. As new records for this student are added and the original area is not enough to hold the new records, the data area grows in multiples of 100 words.

By organizing the SDB file in this way, the data for each student is stored in a variable-length data area which contains current, history, group link, and simple records, and any of these records can be used to hold curriculum or demographic data. To identify whether a record holds curriculum or demographic data (and if curriculum, for which instructional program), code numbers are associated with instructional programs, and demographic data is associated with the code number 0. Table E-3 specifies this code number association (notice that WDRSD is treated as three instructional programs). Table E-4 shows the format for storing current records of the most recent performance data on an instructional module. The format for storing history records of previously assessed performance data is shown in Table E-5. Group link records may be used to construct the ring structures for the groups (either instructional or familial) to which a student belongs. Table E-6 shows the format of a group link record. Simple records may be used to store the list of instructional modules

TABLE E-3

CODE NUMBERS ASSIGNED TO INSTRUCTIONAL PROGRAMS AND DEMOGRAPHIC DATA FOR IDENTIFICATION OF RECORDS

Code #	Association
0	Demographic record
1	Word Attack record of WDRSD
2	Study Skill record of WDRSD
3	Comprehension record of WDRSD
4	DMP record
5	SAPA record
6 . . N	(Undefined as of this time. May be used to distinguish records of other instructional programs which will be added to the WIS-SIM Generalized System at a later date.)

TABLE E-4

FORMAT OF A CURRENT RECORD USED TO STORE THE MOST RECENT PERFORMANCE DATA ON AN INSTRUCTIONAL MODULE

Data Item	Number of Characters	Number of Fields	Content	Remark
KIND	1	1	1	This field identifies the KIND of record. 1 means Current Record.
AREA	1	1	1-N	This field uses the code number to identify the instructional program to which the data in this record belong.
TYPE	1	1	Level number or Topic number or Module number	Content of TYPE depends on content in AREA. If content of AREA is A. 1 to 3, TYPE = (WDRSD) Level number B. 4, TYPE = (DMP) Topic number C. 5, TYPE = (SAPA) Module number
SIZE	3	1	number of words following this in the record	Used to determine the length of the record.
ATTEMPT	2	1	number of assessments made on skills/objectives defined by AREA-TYPE fields.	
MASTERY	1	1	number of skills/objectives mastered	
OBJECTIVE/ SKILL	6	K	Char. 1 = month of last attempt 2 = day of last attempt 3 = year of last attempt 4 = number of attempts on this skill/objective 5 = score of last attempt 6 = ptr. to history (0 means no history)	K = number of skills/objectives of this level/topic/module. The order of this field in the record determines the objective number/skill number to which the data in the field belong.

Table E-4 (cont.)

Scores in the records can be coded as follows:

0 ⇒ NA meaning not assessed on this skill/objective.

1-100 ⇒ Percentile score in the range 0-99

101 ⇒ NM meaning non-mastery

120 ⇒ N meaning need help

140 ⇒ P meaning pass

160 ⇒ NC meaning not covered but interpreted as mastery during prerequisite checking

180 ⇒ M meaning mastery scored in the instructional program indicated in AREA field of the record

180+AREA^1 ⇒ MX meaning mastery scored in the instructional program indicated by the value of AREA^1

200 ⇒ TC meaning teacher certified mastery

220 ⇒ TO meaning teacher observed mastery

Note: AREA^1 is the code number of any instructional program other than the one under which the score is stored. X represents characters W, X, C, D, or S depending on whether AREA^1 = 1, 2, 3, 4, or 5.

For example: 180 in a DMP record (i.e. AREA = 4)

⇒ M scored in a DMP objective

180 in a SAPA record (i.e. AREA = 5)

⇒ M scored in a SAPA objective

but,

185 in a DMP record ⇒ M scored in an equivalent SAPA objective

184 in a SAPA record ⇒ M scored in an equivalent DMP objective

TABLE E-5

FORMAT OF A HISTORY RECORD USED TO STORE PREVIOUSLY ASSESSED PERFORMANCE DATA ON AN INSTRUCTIONAL MODULE

Data Item	Number of Characters	Number of Fields	Content	Remark
KIND	1	1	2	This field identifies the KIND of a record. 2 means historical record.
AREA	1	1	1-N	This field uses the code number to identify the instructional program to which the data in this record belong.
TYPE	1	1	See TYPE of current record.	
SIZE	3	1	Number of words following this in the record.	Used to determine the length of this record.
OBJECTIVE/SKILL	6	K	Char. 1 = month of this assessment 2 = day of this assessment 3 = last 2 digits of year of this assessment 4 = 0 5 = score 6 = ptr. to previous history (0 means no previous history)	K = number of history scores for the level/topic/module. Score is coded according to the scheme used in current record.

TABLE E-6

FORMAT OF A GROUP LINK RECORD USED FOR EITHER INSTRUCTIONAL OR FAMILIAL GROUP

Data Item	Number of Characters	Number of Fields	Content
KIND	1	1	3
GROUP INDEX	2	1	Index number of the group in group index
PREVIOUS STUDENT IN GROUP	3	1	Index number in student index of the previous student in group
NEXT STUDENT IN GROUP	3	1	Index number in student index of the next student in group

recommended for the student on the latest Grouping Recommendation Report, as shown in Table E-7; they also are used to store the goal setting data as shown in Table E-8. Notice that current, history, and simple records may also be used to store demographic data by coding the AREA field of the record key as 0. No example of these demographic records is given here because it is not yet known what kind of data will be stored. Access to a student's records is through the student index number (the student's name and unit or identification number), and then to the student's data area. Access to records of members of a group is through the group index by using the group key. As many students or groups may be included in these indices as are required for the specific application.

TABLE E-7

FORMAT OF A SIMPLE RECORD USED TO STORE THE LIST OF INSTRUCTIONAL MODULES FOR WHICH THE STUDENT IS RECOMMENDED DURING THE LATEST GROUPING RECOMMENDATION REQUEST

Data Item	Number of Characters	Number of Fields	Content
KIND	1	1	4
AREA	1	1	Code number of the instructional program
TYPE	1	1	1
SIZE	3	1	Total number of instructional modules in the instructional program designated by AREA.
ENTRY	1,	Total number of instructional modules in the instructional program designated by AREA.	1, if the student has been recommended for the corresponding instructional module, 0 otherwise.

TABLE E-8

FORMAT OF A SIMPLE RECORD USED TO STORE GOAL-SETTING DATA

Data Item	Number of Characters	Content
KIND	1	4
AREA	1	1-N
TYPE	1	0
SIZE	3	6
Baseline mastery	1	Number of skills/topics/modules mastered at time of inclusion in data base
Current mastery	1	The up-to-date number of skills/topics/modules mastered
1st semester anticipated mastery	1	1st semester anticipated number of skills/topics/modules to be mastered
1st semester actual mastery	1	Actual number of skills/topics/modules mastered in 1st semester
2nd semester anticipated mastery	1	2nd semester anticipated number of skills/topics/modules to be mastered
2nd semester actual mastery	1	Actual number of skills/topics/modules mastered in 2nd semester

APPENDIX F

Functions of a School Interactive Terminal

APPENDIX F

FUNCTIONS OF A SCHOOL INTERACTIVE TERMINAL

KEYBOARD CODE	FUNCTION	INPUT PARAMETERS
IDB	Initiate a Unit Data Base	(School designation established by prior sign-on procedure). Unit designation Student names
UPP	Unit Performance Profile Report	Unit designation Number of copies desired Requested topics
IGR	Instructional Grouping Recommendations	Unit designation Topics desired
IIG.	Implement Instructional Grouping (to accept or modify a previous recommendation)	Unit designation Topic The student numbers from the instructional grouping recommendation that are to be included Names of additional students Teacher's name
GRD	Grading Report (enter student performance data)	Group identification number Students' grades
PDR	Prerequisite Deficiency Report (lists topic prerequisite deficiencies for individual students).	Unit Topic
IPP	Individual Performance Profile	Unit(s) Topic(s) to be covered Student names
ADD	Add a student to a unit	Unit Student name
DEL	Delete a student from a unit	Unit Student name
MOV	Move a student between units	Sending unit Destination unit Student name
MOD	Modify student master record information	Unit designation Student name

KEYBOARD CODE	FUNCTION	INPUT PARAMETERS
RDR	Signals for card reader input	
$$$	Returns control to teletype	
COPY	Immediately reprints previously printed output	
HELP	To provide the terminal user with a set of instructions appropriate to the program in use.	
POS	To request Individual Program of Study Requests	Student number Student name Curriculum area

National Evaluation Committee

Francis S. Chase, Committee Chairman
Emeritus Professor, Department of Education
University of Chicago

Helen Bain
Past President
National Education Association

Lyle E. Bourne, Jr.
Institute for the Study of Intellectual Behavior
University of Colorado

Sue Buel
Dissemination and Installation Services
Northwest Regional Educational Laboratory

Roald Campbell
Emeritus Professor, Department of Educational Administration
The Ohio State University

George E Dickson
College of Education
University of Toledo

L. R. Goulet
Departments of Educational Psychology and Psychology
University of Illinois—Champaign-Urbana

Chester W. Harris
Department of Education
University of California—Santa Barbara

W. G. Katzenmeyer
Department of Education
Duke University

Hugh J. Scott
Department of Education
Howard University

Barbara Thompson
Superintendent of Public Instruction
State of Wisconsin

Joanna Williams
Department of Psychology and Education
Teachers' College, Columbia University

Executive Committee

Richard A Rossmiller, Committee Chairman
Director
R & D Center

William R. Bush
Deputy Director
R & D Center

M Vere DeVault
Professor of Curriculum and Instruction
School of Education

Dale D. Johnson
Assistant Dean
School of Education

Karlyn Kamm
Developmental Specialist
R & D Center

Herbert J. Klausmeier
Principal Investigator
R & D Center

Joel R. Levin
Principal Investigator
R & D Center

James M. Moser
Senior Research Scientist
R & D Center

Len VanEss
Associate Vice Chancellor
University of Wisconsin—Madison

Faculty of Principal Investigators

Vernon L. Allen
Professor
Psychology

B Dean Bowles
Professor
Educational Administration

Marvin J. Fruth
Professor
Educational Administration

John G Harvey
Associate Professor
Mathematics

Frank H. Hooper
Professor
Child and Family Studies

Herbert J. Klausmeier
V A C Henmon Professor
Educational Psychology

Gisela Labouvie-Vief
Assistant Professor
Educational Psychology

Joel R. Levin
Professor
Educational Psychology

Joseph Lins
Professor
Institutional Studies

James Lipham
Professor
Educational Administration

Wayne Otto
Professor
Curriculum and Instruction

Robert Petzold
Professor
Curriculum and Instruction

Thomas A. Romberg
Professor
Curriculum and Instruction

Richard A. Rossmiller
Professor
Educational Administration

Dennis W. Spuck
Assistant Professor
Educational Administration

Richard L. Venezky
Professor
Computer Sciences

Larry M. Wilder
Assistant Professor
Child and Family Studies